Protein Structure Refinement by Optimization

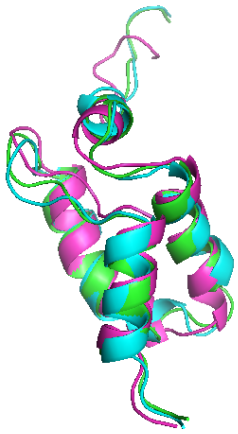
Carlsen, Martin

Publication date:
2016

Document Version
Publisher's PDF, also known as Version of record

TECHNICAL UNIVERSITY OF DENMARK
DEPARTMENT OF APPLIED MATHEMATICS AND COMPUTER SCIENCE

Protein Structure Refinement by Optimization



Ph.D. Thesis

Martin Carlsen

June 2015

Protein Structure Refinement by Optimization

Martin Carlsen

Department of Applied Mathematics and Computer Science

Technical University of Denmark

Title of thesis:
Protein Structure Refinement by Optimization

Ph.D. report number:
PHD-2015-376

ISSN:
0909-3192

Ph.D. student:
Martin Carlsen
Department of Applied Mathematics and Computer Science
Technical University of Denmark
Address: Matematiktorvet, Building 303B, DK-2800 Lyngby, Denmark
E-mail: macar@dtu.dk

Supervisors:
Peter Røgen
Department of Applied Mathematics and Computer Science
Technical University of Denmark
Address: Matematiktorvet, Building 303B, DK-2800 Lyngby, Denmark
E-mail: prog@dtu.dk

Mathias Stolpe
Department of Wind Energy
Technical University of Denmark
Address: Frederiksborgvej 399, Building 115, DK-4000 Roskilde, Denmark
E-mail: matst@dtu.dk

Summary

Proteins are the main active elements of life whose chemical activities regulate cellular activities. A protein is characterized by having a sequence of amino acids and a three dimensional structure. The three-dimensional structure has only been determined experimentally for 50000 of the seven million sequences that are known. Determining the protein structure from its sequence of amino acids is therefore a major problem in computational structural biology and is referred to as the protein folding problem. The folding problem is solved using de novo methods or comparative methods depending on whether the three-dimensional structure of a homologous sequence is known. Whether or not a protein model can be used for industrial purposes depends on the quality of the predicted structure. A model can be used to design a drug when the quality is high.

The overall goal of this project is to assess and improve the quality of a predicted structure. The starting point of this work is a technique called metric training where a knowledge-based protein potential, for a fixed set of native protein structures and a set of deformed decoys for each native structure, is designed to have native-decoy energy gabs that correlates maximally to a native-decoy distance. The main contribution of this thesis is methods developed for analyzing the performance of metrically trained knowledge-based potentials and for optimizing their performance while making them less dependent on the decoy set used to define them. We focus on using the gradient and the Hessian in the analysis and present a novel smooth solvation potential but otherwise the studied potential is kept close to standard coarse grained potentials.

We analyze the importance of the choice of metric both when used in metric training and when used in the evaluation of the performance of the resulting potential and find a significant improvement by using a metric based on intrinsic geometry. It is well-known that energy minimization of a potential that is efficient in ordering a fixed set of decoys need not bring the decoys closer to the native state. The next part of the work is focused on improving the convergence of decoy structures and we present a method that significantly improves the results of shorter energy minimizations of a metrically trained potential and discuss its limitations. In an ideal potential all near-native decoys will converge toward the native structure being at-least a local minimum of the potential. To address how far the current functional form of the potential is from an ideal potential we present two methods for finding the optimal metrically trained potential that simultaneous has a number of native structures as a local minimum. Our results generally indicate that a more fine-grained potential is needed to meet desired model accuracies but even with our coarse-grained model we obtain good results and there is an unexplored possibility to combine it with comparative modeling.

To allow fast energy minimization in Matlab a new set of more sparse formulas to calculate the first and second derivatives of a molecular potential is derived and implemented.

Resumé (in Danish)

Proteiner er de aktive elementer af liv, hvis kemiske aktiviteter regulerer alle celle aktiviviteter. Et protein er karakteriseret ved at have en aminosyrer sekvens og en tre dimensional struktur. Den tre dimensionale strukturer er kun blevet bestemt eksperimentelt for 50000 af de syv millioner sekvenser, der er kendt. Bestemmelsen af protein strukturen fra dets aminosyrer sekvens er derfor en stor udfordring i computationel strukturel biologi og har fået navnet protein foldningsproblemet. Foldningsproblemet løses med de novo metoder eller komparative metoder afhængig af om den tre dimensionale struktur af den homologe sekvens er kendt. Om en model kan anvendes f.eks. til industriel brug afhænger af kvaliteten af den forudsagte struktur. En model kan anvendes til at designe et medikament, hvis kvaliteten er høj.

Det overordnede mål med dette projekt er at vurdere og forbedre kvaliteten af en forudsagt struktur. Projektet tager udgangspunkt i en teknik kaldet metrisk træning, hvor et vidensbaseret protein potential for et sæt af native struktur og et sæt af decoys for hver nativ struktur designes til at have et nativ-decoy energigab, som har en høj korrelation til en nativ-decoy afstand. Hovedbidraget fra denne thesis er metoder udviklet til at analysere ydeevnen af metrisk trænede vidensbaserede potentialer og til at optimere deres ydeevne, samtidig med at de gøres mindre afhængig af det decoy sæt, som anvendes til at definere dem. Vi fokuserer på anvendelsen af gradient og Hessiant i analysen og præsenterer et ny glat solvent potential. Det anvendte potential er baseret på en backbone model af proteinet, men ligner ellers et standard coarse-grained potential.

Vi analyserer vigtigheden af valget af metrik, både når den anvendes i den metriske træning og når den anvendes i evalueringen af ydeevnen af det resulterende potential og finder en signifikant forbedring ved at anvende en metrik baseret på en intrinsisk geometri. Det er velkendt, at energiminimering ved anvendelse af et potential, der er god til at ordne et sæt af decoys, ikke nødvendigvis behøver at forbedre kvaliteten af en decoy. Den næste del af projektet er fokuseret på at forbedre konvergensen af decoy strukturerne, og vi præsenterer en metode, der signifikant forbedrer resultaterne af korte energiminimeringer af et metrisk trænet potential og diskuterer dets begrænsninger. For et ideelt potential vil alle nær-native decoys konvergerer mod en nativ struktur, der er et lokalt minimum for potentialet. For at undersøge hvor langt den nuværende funktionelle form af potentialet er fra et ideelt potential præsenterer vi to metoder til at finde det optimale metrisk trænede potential, som for hver nativ struktur har et lokalt minimum. Vores resultater peger på, at et fine-grained potential kræves for at opnå en høj model nøjagtighed, men selv med en coarse-grained model, opnår vi gode resultater. Endvidere er der en mulighed for at kombinerere potentialet med komparativ modellering.

For at opnå en hurtig energiminimering i Matlab udvikles og implementeres et nyt sæt af formler til at udregne de første og anden afledede af et molekulært potential.

Preface

This thesis is submitted in partial fulfillment of the requirements for obtaining the degree of Ph.D. at the Technical University of Denmark. The Ph.D. project was funded by the Technical University of Denmark and carried out at the Department of Applied Mathematics and Computer Science during the period August 1st 2011 - June 30th 2015. Supervisors on the project were Associate Professor Peter Røgen from the Department of Applied Mathematics and Computer Science and Senior Researcher Mathias Stolpe from the Department of Wind Energy.

List of publications

The following research papers and manuscripts have been written as part of the Ph.D. study

1. M. Carlsen, Using Operators to Expand the Block Matrices Forming the Hessian of a Molecular Potential, *Journal of Computational Chemistry*, vol. 35, pp. 1149-1158, 2014.

2. M. Carlsen, P. Koehl, P. Røgen, On the Importance of the Distance Measures Used to Train and Test Knowledge-Based Potentials for Proteins, *PLOS ONE*, vol. 9, pp. e109335, 2014.

3. M. Carlsen, P. Røgen, Protein Structure Refinement by Optimization, *Proteins: Structure, Function and Bioinformatics*, accepted, 2015.

4. M. Carlsen, P. Røgen, Designing Smooth Knowledge-Based Potentials with Local Minima in Native Structures, in submission, 2015.

During the period of the Ph.D. project I also worked on

5. F. Macchi, S.V. Hoffmann, M. Carlsen, B. Vad, I. Imparato, C. Rischel and D.E. Otzen, Mechanical Stress Affects Glucagon Fibrillation Kinetics and Fibril Structure, *Langmuir*, vol. 27, pp. 12539-12549, 2011.

Acknowledgments

It would not have been possible to write this Ph.D. thesis without the help and support of several people. First of all, I would like to thank my supervisor Peter Røgen for his guidance, assistance and advice. The many fruitful discussions have indeed been appreciated. Many thanks also to Patrice Koehl for making my stay at the University

of California, Davis possible. Finally, loving thanks to my wife Jane whose support has been invaluable to me.

Randers, June 2015

Martin Carlsen

Contents

Introduction

This work is about protein structure prediction which is the prediction of the three dimensional structure of a protein from its amino acid sequence. Proteins fold to a state called the native configuration[1]. The native configuration is predicted with methods such as comparative modelling and de novo methods[2]. The first method predicts structures by finding a homologous amino acid sequence with a known structure in the protein data bank while the second method predicts the folded structure from the sequence alone. Unfortunately, the accuracy of the predicted structures is not always the best possible. The accuracy of a predicted structure is important as the usefulness of a protein structure increases with the accuracy[3, 4]. This work is motivated by the development of knowledge-based potentials for protein structure prediction. Knowledge-based potentials are trained with the purpose of improving the quality of near-native structures. This means either to rank a set of near-native protein structures according to how native-like they are (model quality assessment[5]) or to refine the quality of near-native protein structures (protein structure refinement[6]). Most knowledge-based potentials are stochastic but in this study a knowledge-based potential is a model of the energy landscape of proteins that is trained on a set of near-native structures called decoys. It is usually spanned by basis functions and its free energy parameters are all determined by solving an optimization program. The definition of a knowledge-based potential used here should not be confused with a statistical potential that is based on the frequency of structural motives of protein structures in the protein data bank which also is referred to as a knowledge-based potential.

The quantification of native-likeness is non-trivial as there are many ways to measure the similarity of two structures. We refer to these measures of similarity as distance measures and differ between distance measures based on the extrinsic geometry such as the Root Mean Square Deviation (RMSD)[7, 8, 9] or the Global Distance Test (GDT-TS)[10] and the intrinsic geometry such as the fraction of native contacts (Q) or the energetic difference between the native equilibrium configuration and the perturbed near-native configuration using a fictive spring model (FlexE[11], MT). Distance measures based on the extrinsic geometry are the most popular measures. They measure the distance between two structures after optimal alignment of one structure onto the other. The standard distance measure used is RMSD. On the other hand, distance measures based on the intrinsic geometry require no structural alignment but use a fixed sequence.

This study takes the optimization method introduced in Ref. [12] as a starting point to design a knowledge-based potential by a least square procedure that minimizes the squared difference between the native-decoy energy gab and the native-decoy distance. Its purpose is to form a well-correlated energy landscape about a set of native structures such that the correlation between a knowledge-based potential and a distance measure is high. The knowledge-based potential used in Ref. [12] is quite good at ordering

decoys according to how native-like they are but is unsuitable for improving the quality of near-native protein structures as it does not sustain the local geometry and contains terms that are not differentiable. Differentiability of the constructed potential is important since we want to use a potential energy minimization method for protein structure refinement that requires that the gradient and the Hessian of the potential is available.

The aim of this study is to design a smooth knowledge-based potential and use it for model quality assessment and protein structure refinement. We attempt to approximate the energy landscape using reduced models of proteins and simplified b-spline expanded potentials. The developments of this project are the following:

We first perform a large scale test and analysis of metric training of knowledge-based protein potentials introduced in Ref. [12]. We are interested in finding out the maximal performance that can be attained with a b-spline expanded C-alpha pair distance based knowledge-based potential when trained on different distance measures and tested on a variety of test sets. This knowledge-based potential is smooth and thus has gradient and Hessian which we want to use for for protein structure refinement. The main conclusion of this investigation is that it is best to train the knowledge-based potential on an intrinsic distance measure, namely MT, which thus is our preferred choice in subsequent work. Having established that a single pairwise potential yields high energy-distance correlations when trained on MT and thus is very suitable for model quality assessment we develop a knowledge-based potential for protein structure refinement. This is much more difficult than assessing the quality of model as the local geometry has to be preserved during the optimization. Furthermore, it is very difficult to design a model that consistently improves the quality of a target since we have to ensure that the direction of our refinement procedure is toward the native configuration and not in an arbitrary direction.

We model the protein backbone using five atoms (N, C, C-alpha, O and H) per amino acid in the backbone of the protein. This allows us to model local bonds that restrict motions to be close to a dihedral angle model and also to model main-chain hydrogen bonds. We refer to this potential as the local potential. The non-local part of the potential takes into account the pairwise and solvent interactions. It only depends on the C-alpha atoms and is smooth as it is expanded in terms of cubic b-spline basis functions of the distances between pairs of C-alpha atoms. The full knowledge-based potential developed is thus smooth. The potential may be seen as a smooth version of the most used simple protein models. In the thesis we present some ideas on how to extend the potential with terms that for instance take into account the direction of the side-chains.

The derivatives of the non-local potential are straightforward to derive as they only depend on the inter-residue distances. The formulas for the derivatives of the local potential are, however, non-trivial to derive as the local potential depends on the bond lengths, bond angles and torsion angles also referred to as the internal coordinates. In this thesis I show that there exist simple rules to calculate the derivatives of internal coordinates and thus molecular potentials in general when the formulas are expressed using not one but two orthonormal bases. These formulas are used to calculate the gradient and Hessian of our knowledge-based potential.

We use the developed smooth knowledge-based potential for protein structure refinement using potential energy minimization as our refinement method. We use a modified Newton method that takes into account directions of negative curvature. We only observe improvements in quality of a near-native structure in the beginning of an

optimization. The refinement method that we use thus searches along descent directions for a limited change in RMSD (say $0.5\mathring{A}$) between the initial and final structure. Furthermore, we introduce an iterative method that improves the performance of our smooth knowledge-based potential considerably. We show that the quality of many near-native structures can be improved using this refinement strategy although the improvements admittedly are small.

Finally, a measurement of the gradient and the eigenvalues of the Hessian of our knowledge-based potential at several native structures leads to the conclusion that our knowledge-based potential does not stabilize these native structures. This motivates us to develop two methods to form local minima in the energy landscape. The first is a semidefinite programming approach that achieves this by explicitly requiring a vanishing gradient and a positive semidefinite Hessian simultaneously in a set of native structures. The second method is based on an iterative strategy. Using these methods we conclude that the cost of forming local minima in the energy landscape for a large set of native structures (say larger than 5) is high in the sense that the average correlation between our knowledge-based potential and the distance measure used in the training is low. Hence, the potential looses its desired shape at other folds. This concludes the developments of this project.

The thesis is organized as follows. It is divided into two parts. The first part introduces the background for the thesis as a whole. We begin by describing some of the most important topics in protein structure prediction in Chapter 1. We introduce the protein structure and discuss different coarse-grained models of a protein. Coarse-grained modelling is important as the full structure often is too computationally demanding to use for several applications[13]. The protein folding problem and how to solve it using either comparative modelling or de novo methods are then discussed. Finally, we discuss the problem of assessing and refining the quality of a predicted protein structure which is one of the major challenges in protein structure prediction today. In Chapter 2 we present some of the important properties of b-splines and consider different local and non-local potentials that have been developed during this study. The coupling potential, the local L-DE potential and the side-chain potential considered here are possible extensions to our current knowledge-based potential described above. Different optimization strategies to determine the free parameters of a knowledge-based potential are discussed in Chapter 3. Finally, a modified Newton method is described in Chapter 4.

The second part of this thesis presents four articles that constitute my contribution to the field. The four articles can be found in the Chapters 5, 6, 7 and 8. The main contributions are presented in the following Chapters:

- Formulas for the derivatives of internal coordinates are derived in Chapter 5.
- The importance of the choice of distance measure used to train and test a C-alpha based knowledge-based pairwise potentials is investigated in Chapter 6.
- A knowledge-based with a fixed backbone potential to restrict motions to be close to those of a dihedral model, a fixed hydrogen-bonding potential and a variable b-spline expanded carbon alpha potential consisting of a pair potential and a solvent potential is introduced in Chapter 7.
- An iterative procedure to improve the decoy-convergence of our knowledge-based potential is examined in Chapter 7.

- A method based on semidefinite programming to a stabilize native protein structure is examined in Chapter 8.
- An iterative procedure to stabilize a native protein structure is examined in Chapter 8.

Part I

Background

Chapter 1

Protein structure prediction

1.1 The protein structure

A protein consists of amino acids and there exist 20 different types of amino acids in nature: alanine (Ala), asparagine (Asn), aspartic acid (Asp), cysteine (Cys), glutamine (Gln), glutamine acid (Glu), glycine (Gly), histidine (His), isoleucine (Ile) leucine (Leu), lysine (Lys), methionine (Met), phenylalanine (Phe), proline (Pro), serine (ser), theronine (Thr), tryptophan (Trp), tyrosine (Tyr) and valine(Val). An amino acid in a protein generally contributes with the same atoms to the backbone of the protein namely a nitrogen atom, two carbon atoms, an oxygen atom and two hydrogen atoms (the side-chain of proline is cyclic and its C-alpha atom is linked to the nitrogen atom). The chemical formula for the atoms in the backbone is thus the same for all amino acids. This is shown in Figure 1.1. The N-terminus is at the start of the backbone and the chain is ended with the C-terminus as shown in Figure 1.2 and 1.3. The amino acids differ from each other by the chemical formula for the atoms in the side-chain. The amino acids are divided into three groups: The non-polar amino acid group (Ala, Gly, Ile, Leu, Met , Phe, Pro, Trp, Tyr and Val), the charged polar amino acid group (Arg, Asp, Glu, His and Lys) and the uncharged polar amino acid group (Asn, Cys, Gln, Ser and Thr) denoted NPo, CPo and UPo. The non-polar amino acids are referred to as hydrophobic and the polar amino acids are referred to as hydrophilic. The hydrophobic amino acids are mainly found in the core of a protein whereas the hydrophilic amino acids mainly are found on the surface.

A protein structure is divided into the primary, secondary, tertiary and quaternary structure. The primary structure is the sequence of amino acids. In the backbone a nitrogen atom, a hydrogen atom, a carbon atom and an oxygen atom are found re-

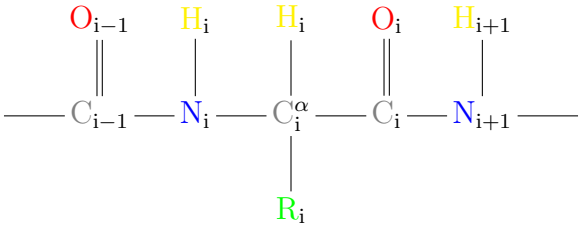


Figure 1.1: Showing the backbone of a protein. H is a hydrogen atom, O is an oxygen atom, N is a nitrogen atom, C is a carbon atom and R is a side-chain. The subindex i refers to the i-th amino acid.

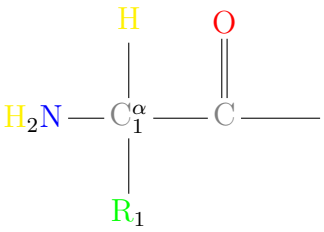


Figure 1.2: The N-terminus (the start of a protein)

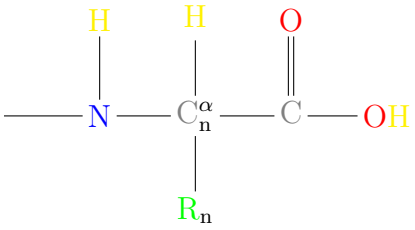
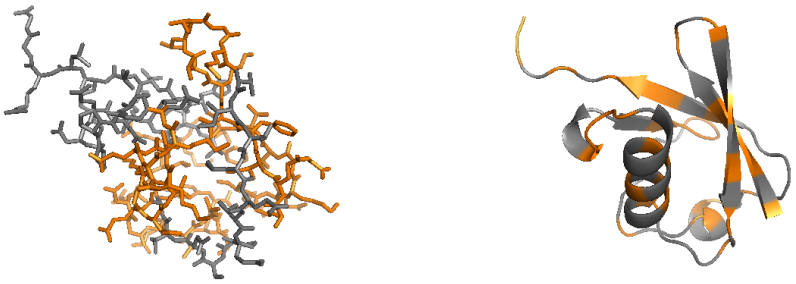


Figure 1.3: The C-terminus (the end of a protein)



(a) (b)

Figure 1.4: Showing the structure of ubiquitin (pdb code 1UBQ). The hydrophobic atoms have been coloured orange and the hydrophilic atoms have been coloured gray. To the left: All of the main-chain and side-chain atoms are shown. To the right: A cartoon showing the secondary structures of the protein.

peatedly in the protein. They function as hydrogen bond donors and acceptors. The hydrogen bonds form important regular structures such as helices and sheets which are structural motives in the protein structure. The structural motives are build up of a hydrogen bond network where the i-th $C = O$ binds to the j-th $N - H$ in the backbone of the protein. For ordinary alpha helices $j = i + 4$. For 3_{10} and π helices $j = i + 3$ and $j = i + 5$, respectively. Beta strands are strands of amino acids where each strand typically is 5 to 10 amino acids long. The individual motives are connected either by loops or turns. They are found primarily on the surface of the protein and thus consist mainly of hydrophilic amino acids or mainly hydrophobic in the case the side-chains of the beta-strands point away from the solvent. The structural motif that the regular and irregular structures form is referred to as the tertiary structure of the protein. The tertiary structure of the protein ubiquitin is shown in Figure 1.4. Finally, the quaternary structure is the combination of protein subunits.

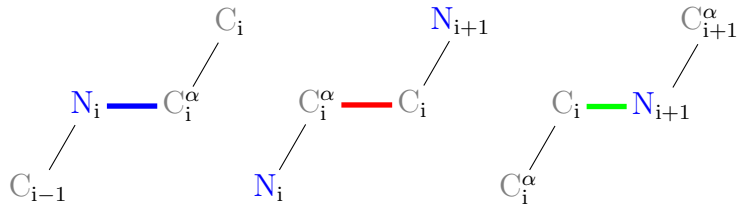


Figure 1.5: The dihedral angles along the backbone of a protein. A dihedral angle is defined for four consecutive atoms. From left to right: the ϕ_i angle , the ψ_i angle and the ω_i angle.

1.2 Amino acid alphabets

Different types of alphabets for amino acids are used in protein structure prediction. In this work, reduced amino acid alphabets are important as it can reduce the number of parameters in a knowledge-based potential. Here, we briefly describe some of the many different types of amino acid alphabets. In the next section, we will consider a reduced amino acid alphabet when we discuss the coupling potential and the local L-DE potential.

The amino acids can be divided into different groups depending on the choice of complexity and size of the sequence. The most simple grouping is when all amino acids are equal and, subsequently, a division into hydrophobic and hydrophilic amino acids or a division into NPo, CPo and UPo. Furthermore, we can divide amino acids into different types of side chains i.e. whether it is an aliphatic side chain, aliphatic hydroxyl side chain, secondary amino group, acidic side chains and their amide derivatives, sulfur containing side chain, basis side chain or aromatic side chain. Furthermore, we can group the amino acids into the secondary group that the amino acid is part of (alpha-helix, beta-sheet or coil). There do however exist statistical methods with the purpose of dividing the amino acids into groups[14]. For a local sequences of N-mer where N is the number of amino acids that are included in the sequence we ask the question: what is the loss of information from choosing a reduced amino acid alphabet. As an example the authors find that a reduction in the amino acid alphabet for a 4-mer from a 20-letter alphabet to a 6-letter alphabet only results in a halving of information. This should be compared with the fact that there exist $20^4 = 160000$ different local 4-mer sequences in the 20-letter alphabet and only 1296 sequences in the 6-letter alphabet.

1.3 Representation and flexibility

For many practical purposes it is not possible to study the all-atom models since the computational time becomes too large[13]. Different levels of generalizations from a single bead per atom to an all-atom representation of an amino acid can be considered depending on application of the protein model. There is a close connection between how we represent a protein and the flexibility of a protein. In the most simple image each amino acid is represented by an atom. Here, the tradition is to choose the C-alpha atom or the C-beta atom (the first carbon atom in the side-chain). This is often sufficient for building the most simple prototypes. Such models are referred to as coarse-grained models. The advantage of using a C-alpha only or a C-beta only model of a protein is obvious. The number of atoms in the model is reduced to the number of amino acids

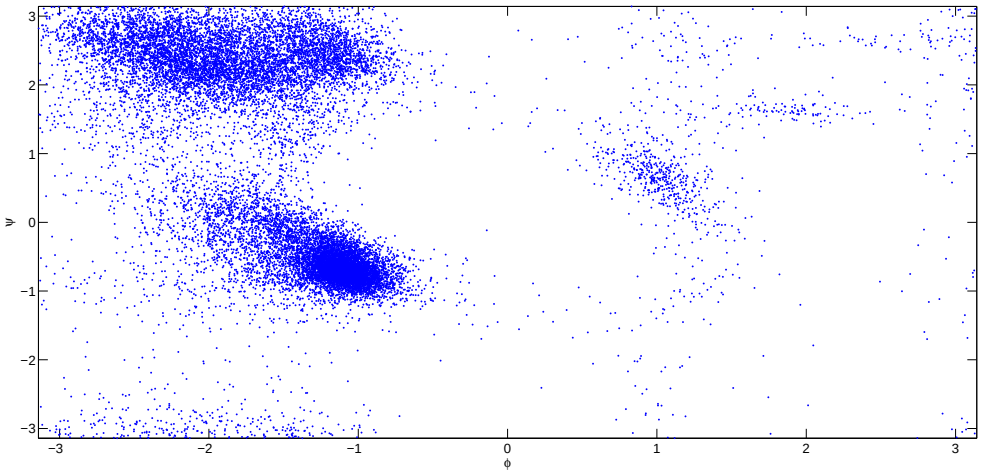


Figure 1.6: A Ramachandran plot of an ensemble of native proteins. Glycine and proline are excluded.

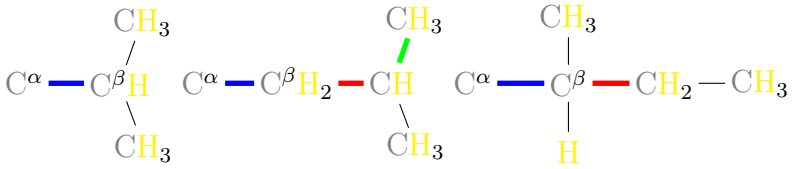


Figure 1.7: Three examples of amino acids with rotameric angles. The first, second and third rotameric angle have been given the colors blue, red and green. From left to right: Valine (Val), Leucine(Leu) and Isoleucine (Ile)

and as a consequence the number of degrees of freedom in the model is small. On the other side, it is not trivial how the hydrogen bonds should be modelled such that the secondary structures are included in the model. Furthermore, the majority of the atoms are completely ignored. A more detailed representation is to use two carbon atoms and a nitrogen atom in the backbone of the protein. Next, the hydrogen bonds can be modelled explicitly such that the amino acid is represented by a carbon atom, a nitrogen atom with an adjacent hydrogen atom and a carbon atom with an adjacent oxygen atom. Each amino acid in this system is in this way represented by five atoms (except proline which is represented by four atoms) and if we do not consider the amino acids at the terminus and proline atoms, then a system with N amino acids has 15N degrees of freedom since there are five atoms with three degrees of freedom for each amino acid.

Whether we choose a coarse-grained model or an all-atom model then it is meaningful to define quantities such as the bond lengths, bond angles and torsion (dihedral) angles. A bond angle is the angle between three atoms. Given four successive atoms the dihedral angle is the angle between the two normal planes spanned by the first three and the last three atoms as shown in Figure 1.5. These definitions are independent of the choice of atoms and can of course be defined for the C-alpha atoms[15], for the backbone atoms or for the side-chain atoms.

When the peptide plane is modelled i.e. when each amino acid is represented by two carbon atoms and a nitrogen atom, then we refer to the rotation angle about the bond

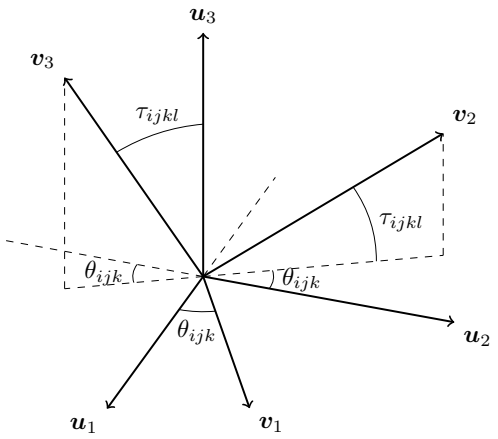


Figure 1.8: Showing how the coordinate system $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3\}$ can be calculated from $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3\}$ by first rotating the coordinate system about $\boldsymbol{u}_1$ using the torsion angle τ_{ijkl}, and thereafter rotating the coordinate system about $\boldsymbol{u}_3$ using the bond angle θ_{ijk}.

between the nitrogen atom and the C-alpha atom and the C-alpha atom and the carbon atom as the ϕ and ψ dihedral angles while the rotation angle about the bond between the carbon atom and the nitrogen atom is referred to as ω. This is illustrated in Figure 1.5. As the dihedral angle ω is a rotation about the peptide bond it is locked at either 180° (trans) or 0 (cis). A 2-dimensional plot of the ϕ and ψ dihedral angles is referred to as a Ramachandran plot and is shown in Figure 1.6 where I have excluded glycine and proline since their conformational flexibility is different from the other amino acids. Due to steric restrictions there are domains of the Ramachandran plot that are forbidden. This means that only a small part of the Ramachandran plot is used which clearly is seen in Figure 1.6. Besides the dihedral angles in the backbone, conformational flexibility is also present in the side-chains. They are referred to as rotameric structures. Some examples are shown in Figure 1.7.

It is possible to introduce the bond angles and torsion angles as Euler rotations. Let $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3\}$ and $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3\}$ be two orthonormal bases. In Figure 1.8 it is shown how the two orthonormal bases are related. It is seen that the bond angle θ_{ijk} and the torsion angle τ_{ijkl} are given by

$$\begin{aligned} \theta_{ijk} &= \cos^{-1}\left(\boldsymbol{u}_1^T \boldsymbol{v}_1\right) \\ \tau_{ijkl} &= \text{sign}\left(\boldsymbol{u}_3^T \boldsymbol{v}_2\right) \cos^{-1}\left(\boldsymbol{u}_3^T \boldsymbol{v}_3\right). \end{aligned} \tag{1.1}$$

This is useful for example to calculate the Cartesian coordinates of the atoms from the internal molecular coordinates: Let $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ be the coordinates of a point $\boldsymbol{q}$ relative to $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3\}$ and $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3\}$, respectively. We consider the coordinate transformation between $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$. Given the point $\boldsymbol{q}_2$, we can find a rigid transformation such that

$$\boldsymbol{q}_1 = \boldsymbol{S}_{12}\boldsymbol{q}_2 + \boldsymbol{q}_{12}, \tag{1.2}$$

where $\boldsymbol{q}_{12}$ is a translation that connects $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ and $\boldsymbol{S}_{12}$ is a rotation matrix. It is easy to see that they only depend on the internal coordinates[16, 17]. Hereby, all the atoms can be transformed to a basis system with a set of rigid transformations.

Finally, there exist other ways to visualize the conformation of a protein. As an example the Gaussian distribution function is used on $SO(3)$ and $\Re^3$ in Ref. [18] to

visualize the distribution data for a protein as an alternative to the Ramachandran plot. Here, they focus on possible positions and orientations between the amino acids which are proximal in space and distal in sequence; the so-called pose coordinates.

1.4 The protein folding problem

The protein folding problem consists in predicting the three-dimensional structure of a protein from its amino acid sequence. The two most important explanations of the protein folding process are based on either 1) that the protein folds hierarchical or 2) that the folding process is driven by the creation of a hydrophobic core. From the hierarchical perspective the protein first forms local structural motives and after that folds to its three dimensional fold. Helix and sheet motives therefore appear first in the folding process. Only then is the hydrophobic core formed which gives rise to the compact structure that a protein folds to. The fact that local motives can be predicted is first and foremost due to steric hindrances in the torsional space which are the cause of local alpha and beta motives that appear with a probability in the torsional space. From the hierarchical perspective the backbone is the key to understand the folding process. It differs from the side-chain centric view where the hydrophobicity of the individual amino acids is the key to understand why a protein folds to a particular structure. It is however also difficult to imagine a model which does not include hydrophobicity. The difference is that from the side-chain perspective it is the hydrophobicity and the compact fold that drive the folding process. See references in Ref. [19] for evidence for the backbone-centered view and the side-chain centered view.

Levinthal's paradox is the immediate paradox that it is almost impossible to find a native structure with a random search in the structural space since the number of degrees of freedom and thus the folds that the protein can fold to is enormous. The probability to find the native structure among all structures is close to zero. The issue with this view is that the configurational space can be biased which is why some configurations have a much larger probability to appear than other configurations. Considering this fact it is not paradoxical that the protein folds to a particular structure since this structure has a large probability. A quantification of this is that the native protein structure is at the bottom of a funnel and the protein fold thus seeks to the bottom of the funnel during the folding.

1.5 Energy landscapes

An energy landscape or an energy surface potential (PES) is a function that depends on a set of coordinates $f(x_1, x_2, \ldots, x_{3n})$ where $x_1, x_2, \ldots, x_{3n}$ are the atomic or molecular coordinates of the molecule. PES is a high dimensional function since the number of the atomic or molecular coordinates in the molecule typically is very large. The assumption that the form of the energy landscape is funnel shaped goes beyond the definition of an energy landscape, as one can easily imagine a landscape with several funnels and thus many possible realisations. The lowest minimum in the energy landscape is referred to as the global minimum and this configuration is the configuration of the native protein. An exponentially high number of local minima can exist above this configuration whose walls can be hindrances during the folding. Among the higher laying states there are intermediate states where the protein is partially folded. This can be an alpha helix or a

beta sheet which has been formed or some of the amino acids which begin to form the hydrophobic core in the structure

The energy landscape of a protein is modelled with a number of different methods which all differ from each other by their complexity. Many force fields such as AMBER, CHARMM, ENCAD or GROMACS are based on semiempirical methods where simple forms of the energy terms are used. These force fields consist of local and non-local terms. The intermolecular weaker contributions to the total potential are hydrogen bonds and other non-covalent bonds such as the Van der Waals force. The Lennard-Jones potential is often used to model the steric repulsions of overlapping electron clouds as a consequence of the Pauli exclusion principle. The local terms consist of harmonic potentials and Fourier terms. Some examples are:

$$\begin{aligned} E_{\text{BL}} &= k_{\text{BL}}(r - r_0)^2 \\ E_{\text{BA}} &= k_{\text{BA}}(\theta - \theta_0)^2 \\ E_{\text{DA}} &= k_{\text{DA}}(1 \pm \cos n\tau), \end{aligned} \tag{1.3}$$

where r, θ_0 and τ are the bond length, bond angle and the dihedral angle, r_0 and θ_0 are the bond length and bond angle for an experimentally determined equilibrium position, n is an integer and k_{BL}, k_{BA} and k_{DA} are constants. These terms obviously depend on which atoms that interact. Higher order terms can be introduced to model the experimental data to a higher accuracy. Furthermore, the harmonic expressions are somewhat unsuitable if θ is close to 0 and 2π. Here trigonometric functions are used instead as these functions are periodic.

1.6 Comparative modelling and de novo methods

The number of known protein sequences is much larger than the number of known structures. About 50000 structures and 7 million sequences are known and the ratio between the two numbers is dropping. Consequently, several computational methods and models have been developed to determine the structure of a protein from its sequence to avoid to use experimental methods such as crystallography and NMR mainly because they are expensive. Instead, the structure is determined in silico from known experimental data. This field is referred to as structural genomics. A study of New York Structural Genomics Consortium has shown that 100 sequences can be modelled for each structure. This high number shows the importance of structural genomics. Two methods are used to model a structure: 1) Comparative modelling and 2) De novo structure prediction[2, 3, 4].

Comparative modelling is based on evolution and works if there for a given sequence is a similar sequence in the sequence space whose structure is known. The similarity of two sequences is referred to as their sequence homology which is quantified with programs such as PSI-BLAST. In comparative modelling the structure is build from a sequence alignment with a template and subsequently the core, loops and chains are build with methods such as rigid-body assembly, segment matching or modeling by satisfaction of spatial constrains. Finally, statistical knowledge-based methods are used to evaluate packing, formation of hydrophobic core, residue and atomic solvent accessibility, spatial distribution of charged groups, distribution of atom-atom contacts and main-chain hydrogen bonding. All in all, there are four steps in comparative modelling:

finding known structures related to the sequence to be modelled (i.e. templates), aligning the sequence to the templates, building a model and assessing the model [20].

A more direct method is to model the physical energy landscape of a protein whose structure is unknown. The methods are referred to as de novo methods as they only require the sequence as input but do not require other structures or sequences to be known. They are also referred to as ab initio methods but since the methods are based on semiempirical potentials whose force fields and parameter sets are estimations and different from model to model the name "de novo" may be more appropriate as it suggests that the methods have been developed to predict new folds. The de novo methods find the global minimum in the energy landscape using a search algorithm. The semiempirical score function and the procedure behind the searching algorithm are thus the crucial elements when developing these methods. An example of a de novo method is Rosetta[21] where the structure is build from local 3-mers and 9-mers of known structures. The scoring function consists of a sequence dependent hydrophobic burial and specific pair interactions such as electrostatics and disulphide bonding and sequence-independent terms representing hard sphere packing, alpha-helix and beta-strand packing, and the collection of beta-strands in beta-sheets. The search algorithm used is simulated annealing.

1.7 Model quality assessment and model refinement

A predicted structure has an accuracy which is determined by how close it is to the experimental structure. The accuracy is most often measured by either RMSD or GDT-TS[1]. The accuracy of the comparative methods is highly dependent on the sequence similarity (SI) between the found template and a target. A distinguishment is made between high resolution ($SI \geq 50\%$, $RMSD \leq 1\mathring{A}$), medium resolution ($30\% \leq SI \leq 50\%$, $RMSD \leq 1.5\mathring{A}$) and low resolution ($SI \leq 30\%$, $2\mathring{A} \leq RMSD \leq 8\mathring{A}$) structures. De novo methods predict structures with $4\mathring{A} \leq RMSD \leq 8\mathring{A}$ i.e. structures with low resolution. We remark, that the definition of low, medium and high resolution often

[1]RMSD is the Euclidean distance between two structures after one of the two sets of atoms $\{a_i\}$ and $\{b_i\}$ has been optimally transformed by a rigid body transformation G:

$$RMSD = \min_G \sqrt{\frac{\sum_{i=1}^{N} \|a_i - G(b_i)\|^2}{N}}. \tag{1.4}$$

The rigid body transformation G is a transformation that does not produce changes in the size, shape, or topology of the protein. Such transformations are compositions of rotations and translations. An issue with RMSD is that it is highly sensitive to outliers, for example due to the presence of large albeit local differences between the two structures such as misorientations of tails and loops. The global distance test (GDT) was developed to decrease this sensitivity[10]. GDT focuses on the regions of the structures that can be correctly aligned by counting the number of residues that can be superimposed within a given cutoff distance. GDT-TS (where TS stands for Total Score), combines this information for multiple cutoffs:

$$GDT - TS = \frac{n_1 + n_2 + n_4 + n_8}{4n}, \tag{1.5}$$

where n_1, n_2, n_4, and n_8 are the numbers of aligned residues within 1, 2, 4, and 8 Ångströms, respectively, and n is the total aligned length. Note that GDT-TS is a quantity between 0 and 1 that represents similarity, with low values corresponding to bad correspondences, and high values (close to or equal to 1) indicating that the two models are highly similar.

depend on the context. The primary causes for the low accuracy are mistakes in side-chain packing, core distortions or loop modelling. If the error in RMSD is greater than say $4\mathring{A}$ then it is possible that model does not have the correct topology. As RMSD is not accurate at this distance, $GDT - TS < 0.5$ (or the TM-score[22]) is used instead to indicate that two structures have a different topology.

The usefulness of a protein structure depends on how accurate the structure is determined. Design and screening of drugs, ligand docking and molecular replacement are possible applications for structures that have a high or medium resolution. Structures with a low resolution have a more modest usefulness such as protein domain boundary identification, topology recognition and family/superfamiliy assignment (see Figure 1 in [3] and [4]). For application in biological research it is thus crucial that the accuracy of a structure is the best possible.

CASP (Critical assessment of structure prediction) is a biannual experiment where the participants are asked to make blind predictions of structures. The competitors either use comparative modelling or de novo methods giving rise to two categories: the template based modelling category (TBM) and the free modelling category (FM). The most successful structure prediction methods are currently I-TASSER[23, 24] and Rosetta[25]. Besides the two categories there are several other categories of which we will consider the model quality assessment category and the model refinement category. The two categories aim at improving the quality of template-based or template-free models so that they can be used for applications such as drug design.

Every second year the performance of the best quality estimation methods are evaluated in CASP[5]. The purpose of this category is to test the current state-of-the-art methods for their ability to give an absolute estimate of the quality of an ensemble of decoys (near-native configurations). This is useful as an ideal score method would allow an estimation of for instance RMSD or GDT-TS without knowledge of the native structure. Currently, the most successful methods are consensus methods. As opposed to single model methods these methods use information from the ensemble of structures to rank the individual models. For all of the methods, the assessors of this experiment find that a large quality range of the ensemble is crucial to the success of these methods.

The predictors in CASP are also tested for their ability to refine models[6]. The purpose of this category is to develop methods that can draw models closer to the native structure and thereby improve the accuracy of the models. There have been developed several methods with the goal of improving the quality of near-native structures. They are usually divided into two groups. Those that are based on potential energy minimization (PEM) and those that are based on molecular dynamics (MD). One of the best performing groups that use PEM is KB01 (or KnowMIN) which is based on the ENCAD potential where non-bonded interactions in the potentials have been replaced with a smoothing of the statistical all-atom potential, RAPDF[26]. 01 refers to the width of the bins ($0.1\mathring{A}$) that has been used in the underlying statistical potential. This potential is also referred to as a hybrid potential as the local potential comes from a MD-potential whereas the global potential is a statistical potential. The newest hybrid potentials sustain the hydrogen bonds and make stereochemical corrections[27, 28]. A study of long molecular dynamics simulation of $100\mu s$ of the CASP 8 and CASP 9 refinement targets using the CHARMM22 force field has shown that a molecular potential of sufficiently high accuracy can refine structures with molecular dynamics despite that their results were limited[29]. In particular, better results were observed when they introduced harmonic restrains on all of the C_α atoms in the secondary structures. Furthermore,

even better results were achieved when they in addition to the restrains on the MD-simulations also introduced an ensemble average[30, 31].

Another refinement strategy is to combine a quality assessment method with a sampling method that generates ensembles of decoys close to the native state [32]. The procedure is iterative and shifts between a generation of near-native decoys using a sampling scheme and an estimation of which one of the decoys that is closest to the native state. When comparing three different sampling techniques: lattice-based coarse-grained sampling, very short all-atom molecular dynamics simulations in implicit solvent and extrapolation of normal modes, the most effective sampling strategy is to use normal mode analysis[32]. Furthermore, the convergence of the iterative procedure is only possible when the size of the random noise in the force fields does not exceed some level[33]. The work, therefore, suggests that a scoring function has to be highly correlated to RMSD (or a similar measure of distance) when using stochastic optimization for it to be useful for protein structure refinement.

The released models in the CASP refinement category are usually close to the native-structure. The predictors are tested for their ability to improve backbone conformation, side-chain packing and local geometry the most important being the backbone conformation. In CASP 10 the backbone conformation was improved in almost 90% of models for the best groups although the improvements were only modest. This shows that the top groups are able to consistently improve the models. The majority of the groups, however, did not draw the model closer to the native structure and no groups were able to produce a model that is closer to the native structure than to the starting model. All in all, the assessors of this experiment find that currently the most successful method is a molecular dynamics method.

CASP is important for this study as we in the beginning of this project hoped that our method would be able to compete with the best methods in CASP in the assessment and the refinement category. Our data, however, suggested that the performance of our methods was good but that our methods could not compete with the best methods in CASP. Focus during this study has thus been how we could improve the performance of our methods. In paper II we investigate which distance measure that is best to use in a metric training of a pair potential (both as a single-model method and as a consensus method) and compare the performance to two other methods (RAPDF and GOAP). In paper III we develop an iterative method to improve the decoy-convergence of our metrically trained knowledge-based potential. Both studies result in an improved performance of our knowledge-based potential but the methods have to be improved further if they are to be successfully used in CASP. One way to do this is to extend our knowledge-based potential with new terms that take into account for instance the direction of the side-chains. In the next chapter we present the different knowledge-based potentials that have been developed in this study.

Chapter 2

B-splines and knowledge-based potentials

In this chapter some of the models that have been developed in this study are presented. Many of these models have been used in the articles that the study has given rise to. We use b-spline functions to span the functional space. This is because they have preferable mathematical properties such as:

1. B-splines are piecewise polynomials of degree p
2. B-splines are non-negative
3. B-splines are compactly supported
4. B-splines form a partition of unity

B-splines are defined as piecewise polynomials of degree p where p is a positive number. Cubic b-splines are often used i.e. b-splines of degree 3. The first property then means that the third derivatives of the basis functions are piecewise constant functions. Furthermore, the first derivatives of the b-spline basis functions are quadratic functions and the second derivatives are continuously piecewise linear functions. B-splines are defined using a knot vector. A knot vector is a sequence of numbers:

$$t_1 \leq \ldots \leq t_{p+1} < t_{p+2} \leq \ldots \leq t_n < t_{n+1} \leq \ldots \leq t_{n+p+1}, \tag{2.1}$$

that each is referred to as a knot. A knot is said to have multiplicity ν if $t_{r-1} < t_r = \ldots = t_{r+\nu-1} < t_{r+\nu}$. Finally, a b-spline is said to be uniform if $t_1 < t_2 < \ldots < t_{n+p} < t_{n+p+1}$ where $t_i - t_{i-1} = constant$ for all i. B-splines with uniform knots are referred to as uniform b-splines. The advantage of using this basis is first and foremost that we can control the knots and thereby where it has its support. It is thus possible to define a potential within a region where data is found and let it vanish for regions where data is not observed and thereby avoid to parametrize regions where the parameters are known to be zero. The definition of the basis functions is:

$$N_i^0 = \begin{cases} 1 \text{ if } t_i \leq t < t_{i+1} \\ 0 \text{ otherwise}, \end{cases} \tag{2.2}$$

and b-splines of degree higher than 0 are found recursively using the formula:

$$N_i^k = \frac{t - t_i}{t_{i+k} - t_i} N_i^{k-1} + \frac{t_{i+k+1} - t}{t_{i+k+1} - t_{i+1}} N_{i+1}^{k-1}, \tag{2.3}$$

for $1 \leq k \leq p$, $i = 1, \ldots, n + p - k$. We use B_i for uniform cubic b-splines. For more information about b-splines, see Ref. [34].

We use two types of b-splines: curves and surfaces. A curve is defined by

$$P(t) = \sum_{i=0}^{n} \alpha_i B_i(t), \tag{2.4}$$

where n is the number of parameters. There are thus $n + 4$ knots for a uniform cubic b-spline with n parameters. A surface is defined by

$$P(s,t) = \sum_{i=0}^{n} \sum_{j=0}^{m} \alpha_{i,j} B_i(s) B_j(t). \tag{2.5}$$

A number of different functional forms can be modelled with b-splines. Due to the great flexibility of these basis functions, uniform cubic b-splines are the fundamental basis that we use to span our knowledge based potentials. Finally, we remark that the derivatives are straight forward to calculate as a knowledge-based potential is linear in its basis functions. Next, we present some of the knowledge-based potentials that have been developed during this study. The pair potential is used in paper II, III and IV and the solvent potential in paper III and IV. The coupling potential, the side-chain potential and the L-DE potential are potentials that we consider as possible extensions to improve the performance of our current model presented in paper III. The purpose of the backbone potential is to sustain the local geometry when refining the structure of a protein. It is used in paper III and IV.

2.1 The pair potential and the coupling potential

One of the most important potentials is the C-alpha pair potential[35, 36, 37]. The idea behind the pair potential is to span a potential in terms of different types of interactions. For each of these a basis is chosen. In the most simple case the basis functions are contact potentials which are turned on when the distance between two amino acids is within an interval. Explicitly, the potential is defined as

$$E_{Pair} = \sum_{i<j} \phi(r_{i,j}), \tag{2.6}$$

where $r_{i,j}$ is the distance between the i-th and j-th interaction pair and

$$\phi(r_{i,j}) = \begin{cases} \alpha_{aa(i),aa(j)} \text{ if } r_{min} < r_{i,j} < r_{max} \\ 0 \text{ otherwise,} \end{cases} \tag{2.7}$$

where aa$(i) \in \{1, \ldots, 20\}$ is the amino acid type of the i-th residue and $\alpha_{aa(i),aa(j)}$ are the parameters. The size of the matrix $\alpha_{aa(i),aa(j)}$ depends on which amino acid alphabet we choose. This can for example be a 20×20 matrix using a 20-letter amino acid alphabet or a 2×2 matrix using a 2-letter amino acid alphabet (hydrophilic and hydrophobic).

The potential well of a contact potential can be modelled more accurately if we choose a more realistic potential such as the Lennard-Jones potential:

$$\phi(r_{ij}) = \frac{\alpha_{aa(i),aa(j)}}{r_{i,j}^{12}} - \frac{\beta_{aa(i),aa(j)}}{r_{i,j}^{6}}. \tag{2.8}$$

Finally, we can model the potential between r_{min} and r_{max} freely. Any potential can in principle be expanded in terms of basis functions :

$$\phi(r_{i,j}) = \sum_k \alpha_k f_k(r_{i,j}), \tag{2.9}$$

where $\{\alpha_k\}$ is a parameter set and $\{f_k\}$ a basis set. If we choose a good basis it means that $\phi(r_{i,j})$ can be spanned using only a few parameters. We choose b-splines as our basis functions:

$$E_{\text{Pair}} = \sum_{i<j} \sum_p \alpha_p^{aa(i)\otimes aa(j)} B_p(r_{i,j}), \tag{2.10}$$

where $B_p(r_{i,j})$ is the p-th b-spline basis function being a function of the distance between the i-th and j-th residues. This pair potential is used in paper II, III and IV.

A possible extension to the pair potential is the coupling potential. The pair potential is dependent on the distance between the atoms but does not couple these movements. We therefore ask the question whether the potential can register whether the i-th atom and the j-th atom is coupled to the j-th atom and the k-th atom. For a pair potential we assume that this is separable into a sum of two terms however if this is not the case then the alternative is a tensor product instead:

$$E_{\text{Coupling}} = \sum_{i<j,j<k} \sum_{p,q} \alpha_{p,q}^{aa(i)\otimes aa(j)\otimes aa(k)} B_p(r_{i,j}) B_q(r_{j,k}). \tag{2.11}$$

It is not hard to see that the number of parameters is huge if we do not choose a reduced amino acid alphabet. A good choice will therefore be an amino acid alphabet consisting of hydrophobic and hydrophilic atoms or an amino acid alphabet based on the type of secondary structure.

2.2 The solvent potential

The pair potential is not by it self able to stabilize the structure of a protein. This is due to the hydrophobic effect that is responsible for the compact form of the protein. The hydrophobic effect is traditionally modelled as the solvent-accessible surface where the accessibility is proportional to surface area[38]. A method for calculating the solvation free energy based on the solvent-accessible surface has been introduced in Ref. [39]. As this solvent potential unfortunately is not differentiable we consider a simple b-spline model of the number of contacts within an sphere. It seems intuitive that such a model would correspond to a model based on the solvent-accessible surface since we expect that the number of contacts is greater for hydrophobic amino acids than for hydrophilic amino acids.

The solvent potential has the form:

$$E_{\text{Solv.}} = \sum_i \sum_p \alpha_p^{aa(i)} B_p(\sum_j t(r_{i,j})),$$

where the basis functions depend on the number of neighbors, $\sum_k t(r_{j,k})$ and t has a value between 0 and 1 in the interval between $0\mathring{A}$ and $10\mathring{A}$ as shown in Figure 2.1. To ensure that the solvent potential is differentiable, we require that t is differentiable. This is fulfilled by defining t as a sum of b-splines basis functions and exploit that they form a partition of unity (see Figure 2.1). We use the solvent potential in paper III and IV.

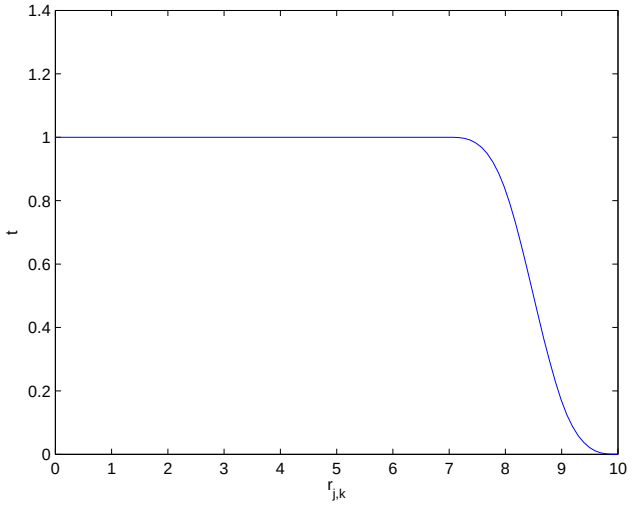


Figure 2.1: Showing the function t which is a smoothed version of a contact between two C_α atoms.

2.3 The side chain potential

We consider a side-chain potential as an add-on to an existing C-alpha pair potential. It is reasonable to conclude that indirectly it is included in the C-alpha pair potential with 20 different amino acids as the amino acids only differ by their side-chain. The pair potential is however not directional dependent and the atoms in the side chains are completely ignored. We want to model this directional dependence of the side-chains without having to introduce an all-atom model i.e. a pair potential defined for all the atoms. We did this to reduce the number of parameters in the model. Unfortunately, we did not have time to implement this model but it may be used to improve the performance of the potential presented in paper III.

Let $\boldsymbol{p}_i$ be the coordinates of the i-th C-alpha atom and define $\boldsymbol{v}_i$ as the unit vector from the i-th C-alpha atom to the center of mass of the side-chain. Next, we consider

$$\begin{aligned} \cos\phi_{ij} &= \frac{\boldsymbol{p}_j - \boldsymbol{p}_i}{\|\boldsymbol{p}_j - \boldsymbol{p}_i\|} \cdot \boldsymbol{v}_i \\ \cos\phi_{ji} &= \frac{\boldsymbol{p}_i - \boldsymbol{p}_j}{\|\boldsymbol{p}_i - \boldsymbol{p}_j\|} \cdot \boldsymbol{v}_j. \end{aligned} \tag{2.12}$$

When $\cos\phi_{ij} = 1$ then $\boldsymbol{p}_j - \boldsymbol{p}_i$ and $\boldsymbol{v}_i$ are parallel. This means that the vector from the i-th C-alpha atom to the j-th C-alpha atom is parallel to the unit vector that indicates the direction of the i-th side-chain. Let

$$t_{ij} = \frac{1 + \cos\phi_{ij}}{2}. \tag{2.13}$$

We remark that if $t_{ij} = 1$ and $t_{ji} = 1$ then the side-chains point towards each other and away from each other if $t_{ij} = 0$ and $t_{ji} = 0$. Finally, if $t_{ij} = 1$ and $t_{ji} = 0$ or $t_{ij} = 0$ and $t_{ji} = 1$ then the side-chains point in same direction. The side-chain potential is given by

the convex combination:

$$\begin{aligned} E_{CH}^{ij} =& t_{ji}t_{ij}\sum_p \alpha_p^{aa_i\otimes aa_j} B_p(d_{ij}) + t_{ji}(1-t_{ij})\sum_p \beta_p^{aa_i} B_p(d_{ij}) \\ &+ (1-t_{ji})t_{ij}\sum_p \gamma_p^{aa_j} B_p(d_{ij}) + (1-t_{ji})(1-t_{ij})\sum_p \delta_p B_p(d_{ij}), \end{aligned} \tag{2.14}$$

where d_{ij} is the distance between the center of mass of the side-chains. The first term has 210 combinations, the second and third term 20 combinations and the last term is side-chain independent. Instead of choosing a convex combination, we alternatively formulate the potential as a tensor product model:

$$\begin{aligned} E_{CH}^{ij} =& \sum_{p=1}^{4}\sum_{q=1}^{4} \alpha_{p,q}^{aa_i\otimes aa_j} S_p(\phi_{ij})S_q(\phi_{ji}) + \sum_{p=1}^{4}\sum_{q=5}^{8} \beta_{p,q}^{aa_i} S_p(\phi_{ij})S_q(\phi_{ji}) \\ &\sum_{p=5}^{8}\sum_{q=1}^{4} \gamma_{p,q}^{aa_j} S_p(\phi_{ij})S_q(\phi_{ji}) + \sum_{p=5}^{8}\sum_{q=5}^{8} \delta_{p,q} S_p(\phi_{ij})S_q(\phi_{ji}). \end{aligned} \tag{2.15}$$

As this potential is only dependent on the angles, it should be turned of with for instance a sigmoidal function when the distance between the side-chains goes beyond a fixed limit.

2.4 The local L-DE potential

A local 7-mer potential is introduced in Ref. [12]. The purpose of introducing this potential is to model the configurational space for 7-mers[40, 41] where each 7-mer is defined on the L-DE plane (see below). L-DE stands for length and distance excess and is defined for a 7-mer. By measuring the length and distance excess it is possible to determine whether the 7-mer primarily is part of an alpha-helix, beta-sheet or coil. This potential is non-vanishing when a 7-mer can be found in the CATH database. A strategy based on 12 different amino acid alphabets[14] is thus introduced to ensure that the potential gives a non-vanishing contribution when the 7-mer is not found in the database. This strategy does not work as well as when the 7-mer is found in the database. This motivated us to develop a potential that is modelled on the L-DE plane using b-splines and based on the 3 amino acid alphabet and thus is not dependent on whether the 7-mer can be found in a database. This potential should be considered as a possible extension to our current potential presented in paper III that may improve the performance.

In the following, the L-DE coordinates and the potential based on the L-DE coordinates and spanned by b-splines are described.

The idea is to smooth out the backbone of a C-alpha model of protein and thereafter measure the length and curvature of the smoothed coordinates. We are then capable of differing between the secondary structures alpha-helix, beta-sheets and coils since they have different length and curvature. Consider the coordinate transformation from $\boldsymbol{x}$ which is $3N$ dimensional consisting of N C-alpha atoms and $\boldsymbol{y}$ which is $3N-12$ dimensional:

$$\boldsymbol{y} = J\boldsymbol{x}, \tag{2.16}$$

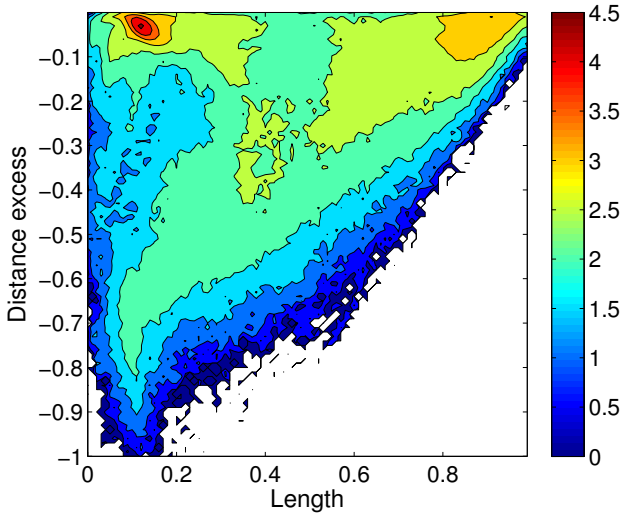


Figure 2.2: Showing the distribution of L-DE coordinates as a logarithmic surface plot. The majority of the data is located at the top left and top right corner where the distance excess is low. The difference between alpha-helices and beta-sheets are thus given by the length coordinate.

where J is a $3N-12 \times 3N$ matrix given by

$$J = \frac{1}{1+2a+2b}\begin{pmatrix} 1 & 0 & 0 & a & 0 & 0 & b & 0 & 0 & a & 0 & 0 & 1 & 0 & 0 & 0 & 0 & \dots \\ 0 & 1 & 0 & 0 & a & 0 & 0 & b & 0 & 0 & a & 0 & 0 & 1 & 0 & 0 & 0 & \dots \\ 0 & 0 & 1 & 0 & 0 & a & 0 & 0 & b & 0 & 0 & a & 0 & 0 & 1 & 0 & 0 & \dots \\ 0 & 0 & 0 & 1 & 0 & 0 & a & 0 & 0 & b & 0 & 0 & a & 0 & 0 & 1 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}, \quad (2.17)$$

and $a = 2.1$, $b = 2.4$. The result of this transformation is a smoothed version of the C-alpha coordinates where a and b have been chosen such that the smoothing of the coordinates is as large as possible. Now, we define the coordinates

$$X_i = s_{i-1,i} + s_{i,i+1} \quad (2.18)$$

$$Y_i = s_{i-1,i+1} - X_i, \quad (2.19)$$

where $\boldsymbol{s}_{i,j} = \boldsymbol{y}_i - \boldsymbol{y}_j$ and $s_{i,j} = \|\boldsymbol{s}_{i,j}\|$ and X and Y such that they are treated equally:

$$X_i' = \beta_X \left(X_i - 2.5\right), \beta_X = \frac{1}{7-2.5} \quad (2.20)$$

$$Y_i' = \beta_Y Y_i, \beta_Y = \frac{1}{0.55}. \quad (2.21)$$

These coordinates are referred to as L-DE coordinates. The distribution of the L-DE coordinates can be found in Figure 2.2. They are divided into three groups. Those that have a distance excess close to zero and a small length, those that have a low distance excess and a long length and the remaining part. The domains correspond to 7-mer fragments of alpha helices, beta sheets and loop domains in a protein structure.

Next, we consider a potential spanned by the tensor products:

$$E_{\text{Local}} = \sum_i \sum_{p,q} \alpha_i^{p,q} B_p(X_i') B_q(Y_i'). \quad (2.22)$$

As for the coupling potential it is preferable to choose a reduced amino acid alphabet since the number of parameters for tensor product splines is quite large. As an example, the potential may be based on a reduced 3-alphabet. In the 3-alphabet the amino acids are divided into the following groups:

1. Asn, Asp, Gly, Pro, Ser and Thr
2. Cys, Ile, Leu, Met, Phe, Trp, Tyr and Val
3. Ala, Arg, Gln, Glu, His and Lys.

This amounts to $3^7 = 2187$ different classes which still are far too many possibilities. If we use 8 degrees of freedom in both the length and the distance excess direction then there are $8 \times 8 \times 3^7 = 139968$ parameters which obviously are too many. Instead we consider a reduced 3-alphabet where we group each 7-mer into how many of each amino acid type the 7-mer contains. We use the notation (a, b, c) for a number of type 1, b number of type 2 and c number of type 3 where $a + b + c = 7$. The distribution of 7-mers when divided into these classes is shown in Figure 2.3. The data points fall into subdomains of the L-DE-plane. For many of the groups this motivate us to define the L-DE potential on subdomains of the L-DE plane thereby reducing the number of parameters in the model even further.

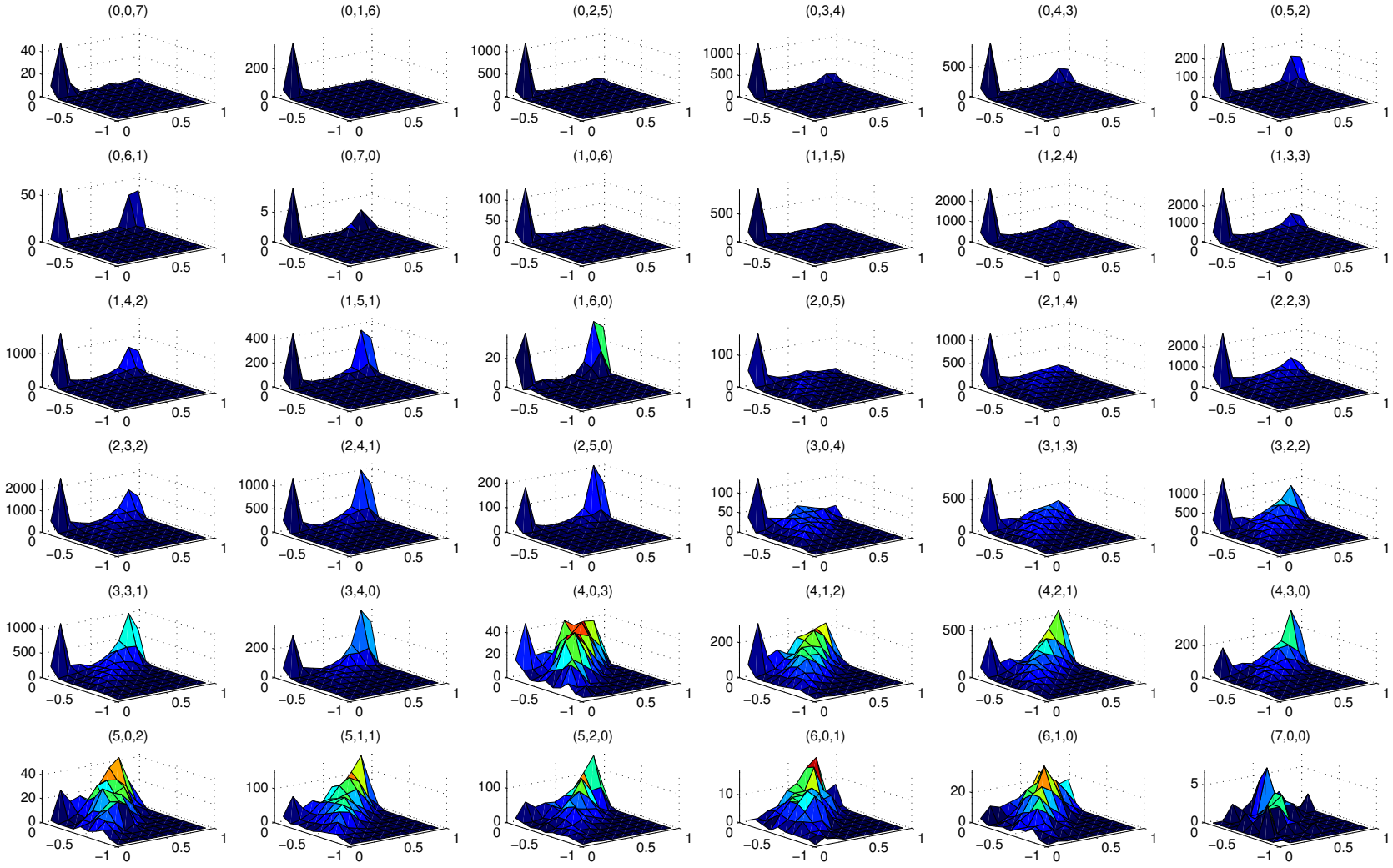


Figure 2.3: Showing the distribution of L-DE points for each of the 36 different groups. The data was obtained for 1174 non-homologous native structures.

2.5 The backbone potential

It is important to sustain the local geometry when we fold a protein. There are several methods to do this. The most simple Cartesian model is typically based on the C-alpha or C-beta atoms with artificial covalent bonds between the atoms. A refinement of this model is a dihedral angle model in Cartesian coordinates where the backbone is sustained by adding harmonic restraints to the bond lengths, bond angles and torsion angles about the peptide bonds. Such a backbone model is intended to mimic a dihedral angle model but with use of Cartesian coordinates. It is sufficient to use three atoms in the backbone of a protein: N, C-alpha, and C.

A geometrical approach to fix the parameters of this potential is to define the bond length potential, the bond angle potential and the proper and (if necessary) improper torsion angle potential by:

$$\begin{aligned}
E_{BL} &= \frac{1}{N_{T_{BL}}}\sum_{i=1}^{M_{BL}} R_i \sum_{k=1}^{N_i^{BL}} \left(r_i^k - \langle r_i\rangle\right)^2 \\
E_{BA} &= \frac{1}{N_{T_{BA}}}\sum_{i=1}^{M_{BA}} \Theta_i \sum_{k=1}^{N_i^{BA}} \left(\theta_i^k - \langle \theta_i\rangle\right)^2 \\
E_{TA} &= \frac{1}{N_{T_{TA}}}\sum_{i=1}^{M_{TA}} T_i \sum_{k=1}^{N_i^{TA}} \left(\tau_i^k - \langle \tau_i\rangle\right)^2 \\
E_{TAIP} &= \frac{1}{N_{T_{TAIP}}}\sum_{i=1}^{M_{TAIP}} \Omega_i \sum_{k=1}^{N_i^{TAIP}} \left(\omega_i^k - \langle \omega_i\rangle\right)^2,
\end{aligned}$$

where R_i, Θ_i, T_i and Ω_i are constants and $\langle r_i\rangle$, $\langle \theta_i\rangle$, $\langle \tau_i\rangle$ and $\langle \omega_i\rangle$ are average values for the respective internal coordinates. The average values and the parameters are determined from their geometrical variation in a set of native structures and are calculated from a set of randomly chosen native protein structures. The parameters for the potential are defined by

$$\begin{aligned}
R_i &\equiv \frac{N_i^{BL}}{\sum_{k=1}^{N_i^{BL}} \left(r_i^k - \langle r_i\rangle\right)^2} \\
\Theta_i &\equiv \frac{N_i^{BA}}{\sum_{k=1}^{N_i^{BA}} \left(\theta_i^k - \langle \theta_i\rangle\right)^2} \\
T_i &\equiv \frac{N_i^{TA}}{\sum_{k=1}^{N_i^{TA}} \left(\tau_i^k - \langle \tau_i\rangle\right)^2} \\
\Omega_i &\equiv \frac{N_i^{TAIP}}{\sum_{k=1}^{N_i^{TAIP}} \left(\omega_i^k - \langle \omega_i\rangle\right)^2}.
\end{aligned}$$

Note that the expressions above are $1/\sigma^2$ where σ is the standard deviation. When we insert the expressions for R_i, Θ_i, T_i and Ω_i we get for this distribution that

$$E_{BL} = E_{BA} = E_{TA} = E_{IPTA} = 1.$$

Apparently, an ensemble of native or near-native configurations will not have the exact energy 1 but in practise and particularly for small perturbations of native structures then

1 will be the typical scale for the energy. This approach essentially results in a dihedral angle model as the parameters for the harmonic potentials that sustain the backbone of protein are high since the geometrical variation of the bond lengths, bond angles and torsion angles about the peptide bonds is small. These parameters may therefore instead be fixed at arbitrary large values while the ϕ and ψ angles are kept free. We use this strategy in paper III and paper IV.

Chapter 3

Optimization strategies

In this chapter we present the background for the optimization method that we use in this study to determine the parameters of our knowledge-based potential. First, we consider optimization methods based on linear programming and Z-score optimization whose purpose is to create a potential that can select the native structure from a set of structures. Next, we introduce the optimization method that we use in paper II - IV[12]. This optimization method results in a potential that can discriminate between the quality of near-native decoys. This is preferable as the native structure often is not available. Finally, we present two ideas how to sculpt a potential that can fold a protein.

3.1 Linear programming

One of the main challenges has been to design a potential or scoring function that can discriminate a native structure from a set of decoys (near-native configurations). This is due to the fact that a scoring function that can discriminate the correct fold from misfolded structures is useful to find the best template among an ensemble of known structures namely the template with the lowest energy. Furthermore, we require that such a potential has the property that the lowest energy is the energy of a native structure compared to the energy of near-native structures. These properties can be achieved by setting up a linear optimization program[42, 43, 44, 45, 46, 47]. The problem is a feasibility problem where we for a linearly parametrized potential require that a set of decoys has a higher energy than the native structure. We thus for a large set of native-decoy pairs of structures require that

$$\Delta E(\text{decoy}, \text{native}) = E_{\text{decoy}} - E_{\text{native}} > \epsilon, \tag{3.1}$$

where ϵ is a small constant, E_{native} is the energy of the native structure and E_{decoy} is the energy of a decoy matching the native structure. We remark that the inequalities define a set of cuts (hyperplanes) in the parametric space. The system is solved by finding a feasible solution that belongs to the feasible polyhedron. For a sufficiently large training set it can be difficult to find a solution. This is discussed in Ref. [42, 43] where they investigate whether it is possible to train a pair potential to recognize a set of 75 protein ensembles and find that the solution is infeasible. There is thus an upper limit on what can be required of a potential and this is of course dependent on the complexity of the potential, the number of parameters and which class the proteins belong to. The same study has been made for a contact potential as well [48]. They find that the existence of

a solution depends on the definition of the contact potentials (the fixing of the cutoff) and the size of the training set. If the training set included the majority of known proteins then no solution is found.

3.2 Z-score optimization

The feasibility problem permits decoys to have an arbitrary energy as long as the energy is higher than the energy of the native structure. There are thus no restrictions on the energy of the decoys compared to the energy of the native structure. We just require that it is higher but there is no limitation on how high. To go beyond the feasibility problem, we allow decoys to have a lower energy than the energy of the native structure but the funnel about the native structure is modelled at the same time[49, 50, 51, 52, 53, 47]. We thus achieve an energy landscape where the energy gab from the native structure to a set of decoy also called the stability gab is maximized. This is achieved by minimizing the Z-score:

$$Z_{\text{native}} = \frac{E_{\text{native}} - < E_{\text{decoy}} >_{\text{decoy}}}{\sigma_{\text{decoy}}(E_{\text{decoy}})}. \tag{3.2}$$

We remark that it is the average energy of a set of decoys that enters into the expression. This means that high energy structures have a higher weight than structures close to the native structure. In a training set there are often many protein ensembles. To avoid that some protein ensembles carry too much weight in the optimization, we thus minimize the Boltzmann-like weighted average of

$$Z_{\text{ave}} = \frac{\sum_{\text{native}} Z_{\text{native}} \exp(-Z_{\text{native}}/k_B T)}{\sum_{\text{native}} \exp(-Z_{\text{native}}/k_B T)}, \tag{3.3}$$

where k_b is Boltzmann's constant and T is the temperature. The optimization problem is solved by a Monte-Carlo procedure. This is a better strategy than just for example taking the average of all Z-scores since there is a risk that too little weight is given to many protein ensembles. To obtain more influence from the low energy structures we instead of the Z-score consider the overlap between two contact matrices ϕ and ϕ' defined by[54]

$$q(\phi, \phi') = \frac{\sum_{i,j} \phi(r_{i,j})\phi'(r_{i,j})}{\max(\sum_{i,j} \phi(r_{i,j}), \sum_{i,j} \phi'(r_{i,j}))}. \tag{3.4}$$

The desired correlated energy landscape is achieved by maximizing the Boltzmann averaged native overlap:

$$Q = \frac{\sum_{\text{decoy}} q(\phi_{\text{decoy}}, \phi_{\text{native}}) \exp(-E_{\text{decoy}}/k_B T)}{\sum_{\text{decoy}} \exp(-E_{\text{decoy}}/k_B T)}. \tag{3.5}$$

Clearly, when $q \approx 1$ the native structure and a decoy have a large overlap i.e. their contact matrices are almost identical. This means that low energy structures have a higher weight in this program as these give rise to $Q \approx 1$. On the other hand, the problematic decoys are removed that are far from the native structure.

3.3 Maximization of the correlation between energy and RMSD

The optimization strategies presented so far aim at discriminating the native structure from a set of decoys. There is no guarantee that they actually form a funnel-shaped energy landscape since the knowledge-based potentials are tested for their ability to identify a native structure among an ensemble of structures. It is clear that the purpose of these methods is to form a knowledge-based potential where the energy of the native structure is low compared to a set of misconfigurations such that the native structure can be discriminated. Furthermore, the methods depend on a different procedure to generate a set of configurations such as gabless threading which are assigned a score.

It is not clear how the energy landscape of these knowledge-based potentials actually look. The form of the energy landscape for example its roughness and the size of the energy barriers is unclear. It is most likely that the potentials create a golf-course landscape in a random energy landscape. This is not necessarily problematic as they only aim at discriminating the native structure among an ensemble of structures but it is difficult to use these potentials for actual folding experiments.

A potential that has been trained exclusively by linear programming or Z-score maximization is not optimized to yield a high correlation between energy and the RMSD between a native structure and a decoy. As a consequence, we expect this potential to have a golf-course landscape where the potential is particularly good at selecting the native structure but is not able to discriminate between decoys. The correlation between the energy of a set of decoys and their RMSD to the native structure may therefore be low. This is a problem since the native structure often is not available. Instead we are interested in a score function that is able to discriminate between a set of decoys and select the decoy with the highest quality.

To improve the ability of the score function to discriminate between decoys we introduce the correlation in the objective function with the Z-score. This is the strategy behind the optimization of the weights (19 in total) of the potential Touchstone II[55, 56] (or I-TASSER[23, 24] that is an automated prediction server based on Touchstone II). The purpose of this optimization is both to maximize the correlation between the energy of the decoys and their RMSD to the native structure and to maximize the energy gap between the native protein and the ensemble of decoys. They find the solution that minimizes the function:

$$G = G_1 G_2 G_3, \tag{3.6}$$

where G_1, G_2 and G_3 are defined by:

$$\begin{aligned} G_1 =& \frac{1}{1 + \langle \text{corr}(RMSD_{\text{native,decoy}}, E_{\text{decoy}}) \rangle_{\text{native}}} \\ G_2 =& \langle \langle \frac{(RMSD_{\text{native,decoy}} - \eta E_{\text{decoy}} + b_{\text{native}})^2}{RMSD_{\text{native,decoy}}} \rangle_{\text{decoy}} \rangle_{\text{native}} \\ G_3 =& \frac{1}{1 - \langle \frac{E_{\text{native}} - <E_{\text{decoy}}>_{\text{decoy}}}{\sigma_{\text{decoy}}(E_{\text{decoy}})} \rangle_{\text{native}}}. \end{aligned} \tag{3.7}$$

The idea behind the first term, G_1, is to optimize the correlation between RMSD and the energy of the ensemble of decoys. The second term, G_2, acts to minimize the χ^2 between a linear regression and the energy versus RMSD where b_{native} is the individual

intercept for a training protein and η is a proportionality constant that is determined from simulations. In the last term, G_3, the energy gab between a native protein and a set of decoys is optimized using the Z-score. Touchstone II is only trained on decoys with $4\mathring{A} \leq RMSD \leq 10\mathring{A}$. If $RMSD < 4\mathring{A}$ then RMSD is set to $4\mathring{A}$ and when $RMSD > 10$ then RMSD is set to 10. This means that decoys that have a RMSD below $4\mathring{A}$ are treated as having a RMSD of 4 while decoys that have a RMSD above $10\mathring{A}$ are treated as having a RMSD of 10. The decoy set thus consists of decoys with a low resolution.

The MPP potential for metric protein potential is a metric trained potential that was developed with the purpose of discriminating decoys of high and medium resolution [12]. With metric training is meant that the energy gab and RMSD are taken to be the same:

$$E_{\text{decoy}} - E_{\text{native}} = \alpha_{\text{native}} RMSD_{\text{native}}, \tag{3.8}$$

where α is positive proportionality constant. When this equality is satisfied it is clear that the correlation between the energy gab of a set of decoys and their RMSD to the native structure is 1. Furthermore the native structure with the lowest energy clearly has a positive energy gab to the set of of decoys since RMSD is non-negative. This follows from the equality. The optimization problem is formulated as a quadratic problem since we want to avoid a feasibility problem. We thus minimize the sum of squared errors:

$$F = \sum_{\text{native}} \sum_{\text{decoy}} \left(E_{\text{decoy}} - E_{\text{native}} - \alpha_{\text{native}} RMSD_{\text{native,decoy}}\right)^2, \tag{3.9}$$

subject to $0.25 \leq \alpha_{\text{native}} \leq 4$ and $\sum_{\text{native}} \alpha_{\text{native}} = N_{\text{native}}$ where N_{native} is the number of native structures in the training. We remark that F and $G2$ are alike and both formulations aim at maximizing the correlation between energy and RMSD. The advantage of using F instead of G is that the proportionality constant is determined in the optimization. Furthermore, it may be difficult to find a solution to G when the potential has many parameters. Touchstone II has 19 parameters whereas MPP has 1312 parameters in total that have to be determined. The advantage of using equation (3.9) is that the terms are additive and the calculation of the individual terms is thus trivially parallelizable. Overall, it means that it is possible to train a knowledge-based potential with many parameters on a large training set.

The objective function defined by equation (3.9) forms the basis of this work and we use it in Paper II - IV. It is a constrained least square problem which is not difficult to show: We first consider the unconstrained quadratic problem

$$\underset{\boldsymbol{X}}{\text{minimize}} \quad f(\boldsymbol{X}), \tag{3.10}$$

where $\boldsymbol{X}$ is a parameter set and the objective function $f(\boldsymbol{X})$ is a quadratic form

$$f(\boldsymbol{X}) = \boldsymbol{X}^T B \boldsymbol{X} + \boldsymbol{X}^T \boldsymbol{c} + d. \tag{3.11}$$

For simplicity we will assume that B is positive definite. The most simple way to ensure this is to use regularization (see below). The matrix B thus has a Cholesky decomposition, $B = A^T A$, and f can be written as

$$f(\boldsymbol{X}) = (A\boldsymbol{X} - \boldsymbol{b})^T (A\boldsymbol{X} - \boldsymbol{b}) + constant, \tag{3.12}$$

such that $\boldsymbol{c} = -2A^T \boldsymbol{b}$. The last term is not important and the original problem is thus equivalent to an ordinary least square problem

$$\underset{\boldsymbol{X}}{\text{minimize}} \quad \|A\boldsymbol{X} - \boldsymbol{b}\|_2^2. \tag{3.13}$$

We remark that we have that $\boldsymbol{b} = \mathbf{0}$ when the quadratic function is given by equation (3.9) since α_{native} is a parameter in the optimization.

One of the advantages of using quadratic programming is that it is not difficult to find a solution that satisfies certain properties. This is done using regularisation techniques. The most simple use of regularization is Tikhonov regularization which consists in adding the matrix $\delta\|\boldsymbol{X}\|_2^2$ to the objective function where $\delta > 0$ is a constant. This ensures that the parameters are kept small and that $\boldsymbol{B}$ is positive definite. The idea can also be used to ensure that wild fluctuations are avoided. This is done by adding $\mu\boldsymbol{X}^T\boldsymbol{C}\boldsymbol{X}$ to the objective function where $\boldsymbol{C}$ is a block diagonal matrix and each of the block matrices are found by calculating the discontinuities of the third derivatives of the b-spline pair potentials in the knots. Hereby, we use that the third derivatives of the cubic b-spline basis functions are piecewise constant functions.

3.4 Funnel sculpting

A method that combines energy optimization with a stochastic search algorithm is developed in Ref. [57]. The purpose of this method is to design a potential that can fold proteins with a stochastic search algorithm. Hereby, an energy landscape is designed where the height of energy barriers has been taken into account. The method which is iterative works in the following way: First, an optimization is started which sets all parameters to zero. Then the algorithm shifts between a parameter optimization of the potential and a stochastic search algorithm from a random configuration. The search algorithm will get stucked if the energy barriers are too high. These decoys (10 in total) are added to the training set at each iteration and the parameter optimization ensures that potential is smoothed out at the places where the search algorithm got stucked. The algorithm continues until the search algorithm reaches within 4Å in RMSD from the native structure. This method is characterised in that the potential is designed such that a search algorithm can fold a protein while small hilltops are kept.

In paper III an iterative method is developed that is inspired by the method above to improve the quality of near-native structures. The idea behind this method is to sculpt the energy landscape in the neighborhood of a set of native structures such that a deterministic search algorithm from a near-native structure converges towards the native structure. The method shifts between a potential energy minimization routine and a parameter optimization using equation (3.9) where the energy of the generated decoys from the minimization routine is raised such that the decoy convergence in the next run is improved. The minimization method is described in detail in the next chapter.

Chapter 4

Potential energy minimization

In this chapter we consider an energy minimization method based on descent directions and directions of negative curvature. Energy minimization methods are used for many different purposes e.g. to find a local minimum and to investigate the normal modes at this optimum. In this study we use it as our refinement algorithm to improve the quality of near-native structures (paper III). The method is also used to improve the convergence of decoy structures (paper III) and to establish a local minimum in the energy landscape (paper IV). It should be said that we decided to use the trust-region large-scale method in Matlab to solve the unconstrained optimization problem instead of the optimization technique presented here as it appeared to be more stable. The principle, however, of the two optimization methods is the same, namely, to use directions of negative curvature of the Hessian to solve the optimization problem.

The unconstrained problem that we are interested in solving can be written as

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad f(\boldsymbol{x}), \tag{4.1}$$

where f is a potential function and $\boldsymbol{x}$ the Cartesian coordinates that f is dependent on. A function is said to be convex if and only if its Hessian is positive semidefinite i.e. for all $\boldsymbol{x}$, $\nabla^2 f(\boldsymbol{x}) \succeq 0$. A molecular or knowledge-based potential that consists of local and non-local terms is not convex. This means that the Hessian of f is indefinite in the non-convex regions.

When using the traditional Newton's method, it is assumed the domain of potential is strictly convex such that the Hessian always is positive definite. The second order approximation $\hat{f}$ of f at $\boldsymbol{x}_k$ in the direction of $\boldsymbol{v}_k$ is:

$$\hat{f}(\boldsymbol{x}_k + \boldsymbol{v}_k) = f(\boldsymbol{x}_k) + \nabla f(\boldsymbol{x}_k)^T \boldsymbol{v}_k + \frac{1}{2}\boldsymbol{v}_k^T \nabla^2 f(\boldsymbol{x}_k)\boldsymbol{v}_k. \tag{4.2}$$

The vector $\boldsymbol{v}_k$ is referred to as Newton's step when it is a solution to the equation:

$$\nabla^2 f(\boldsymbol{x}_k)\boldsymbol{v}_k = -\nabla f(\boldsymbol{x}_k), \tag{4.3}$$

where we assume that $\nabla^2 f(\boldsymbol{x}_k)$ is positive definite.

We remark that the Hessian of a molecular or knowledge-based potential always is singular as the potential is translational invariant. The equation above thus has to be transformed using an affine transformation, $\boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{b}$ where $\boldsymbol{A}$ is a matrix and $\boldsymbol{b}$ a vector, such that $\nabla^2 f(\boldsymbol{x}_k) = \boldsymbol{A}^T \nabla^2 g(\boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{b})\boldsymbol{A}$ and $\nabla f(\boldsymbol{x}_k) = \boldsymbol{A}^T \nabla g(\boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{b})$

Newton's method is probably not the best method to use here as it requires that the Hessian is positive definite. This can be achieved using an approximate positive

definite Hessian but the method does not ensure that the objective function is decreased at each step. A method that ensures descent directions (a direction $\boldsymbol{v}_k$ that satisfies $\nabla f(\boldsymbol{x}_k)^T \boldsymbol{v}_k < 0$ such that $f(\boldsymbol{x}_k + \boldsymbol{v}_k) < f(\boldsymbol{x}_k)$) and guarantees convergence is the modified Newton's method described in Ref. [58, 59]. The idea behind the method is to find two directions: A direction along positive curvature where the Hessian of the potential is approximated by a positive definite matrix and the direction found by calculating Newton's step for the approximate positive definite Hessian and a direction along negative curvature. Specifically, $\nabla^2 f(\boldsymbol{x}_k)$ is LDL^T factorized:

$$\nabla^2 f(\boldsymbol{x}_k) = \boldsymbol{P}\boldsymbol{L}\boldsymbol{B}\boldsymbol{L}^T\boldsymbol{P}^T, \tag{4.4}$$

where $\boldsymbol{P}$ is a permutation matrix, $\boldsymbol{L}$ is a lower triangular matrix and $\boldsymbol{B}$ is a block diagonal matrix. The decomposition of $\boldsymbol{B}$, $\boldsymbol{B} = \boldsymbol{Q}^T\boldsymbol{D}\boldsymbol{Q}$, is used to build the matrix $\hat{\boldsymbol{B}} = \boldsymbol{Q}^T\hat{\boldsymbol{D}}\boldsymbol{Q}$ where the eigenvalues of $\hat{\boldsymbol{D}}$ are given by

$$\hat{\lambda} i = \max\left\{|\lambda_i|, \epsilon_m n \max_{1\leq j\leq n} |\lambda_i|, \epsilon_m\right\}. \tag{4.5}$$

The direction along positive curvature is then found by solving the equation

$$\boldsymbol{P}\boldsymbol{L}\hat{\boldsymbol{B}}\boldsymbol{L}^T\boldsymbol{P}^T\boldsymbol{s}_k = -\nabla f(\boldsymbol{x}_k). \tag{4.6}$$

The direction $\boldsymbol{d}_k$ along negative curvature i.e. $\boldsymbol{d}_k^T\nabla^2 f(\boldsymbol{x}_k)\boldsymbol{d}_k < 0$ is found by solving the equation

$$\boldsymbol{L}^T\boldsymbol{P}^T\boldsymbol{d}_k = \pm\boldsymbol{z}, \tag{4.7}$$

where $\boldsymbol{z}$ is the eigenvector corresponding to the smallest eigenvalue of $\boldsymbol{B}$ and the sign is given by the inequality $\nabla f(\boldsymbol{x}_k)^T\boldsymbol{d}_k \leq 0$. Finally, the step is calculated as $\boldsymbol{s}_k + \beta\boldsymbol{d}_k$ when the Hessian is indefinite. The constant $\beta \geq 0$ is found by solving the equality $(\boldsymbol{s}_k + \beta\boldsymbol{d}_k)^T \nabla^2 f(\boldsymbol{x}_k)\,(\boldsymbol{s}_k + \beta\boldsymbol{d}_k) = \boldsymbol{d}_k^T\nabla^2 f(\boldsymbol{x}_k)\boldsymbol{d}_k$

Part II

Articles

Chapter 5

Using operators to expand the block matrices forming the Hessian of a molecular potential

M. Carlsen, Using operators to expand the block matrices forming the Hessian of a molecular potential , *Journal of Computational Chemistry*, vol. 35, pp. 1149-1158, 2014.

Abstract. We derive compact expressions of the second-order derivatives of bond length, bond angle and proper and improper torsion angle potentials, in terms of operators represented in two orthonormal bases. Hereby simple rules to generate the Hessian of an internal coordinate or a molecular potential can be formulated. The algorithms we provide can be implemented efficiently in high-level programming languages using vectorization. Finally, the method leads to compact expressions for a second-order expansion of an internal coordinate or a molecular potential.

Keywords: second-order derivatives, Hessian, internal coordinates, operators, molecular potentials

5.1 Introduction

The first- and second-order derivatives of a molecular potential are used for geometry optimization[60], normal mode analysis[61] and molecular dynamics[29, 62]. For each of these applications, it is important that they can be calculated analytically. A molecular potential depends solely on functions of internal coordinates consisting of bond lengths, bond angles and torsion angles. The straight-forward approach to calculate the derivatives of an internal coordinate with respect to the Cartesian coordinates is to write it as a function of Cartesian coordinates and use the chain rule. The expressions have been known for decades, and they are long and cumbersome; in particular for the second derivatives of proper and improper torsion angles. The calculation can be performed using a program for symbolic or automatic differentiation[63] but the method is unsatisfactory when there exist a set of simple rules to perform the same calculation. By representing a molecule in internal coordinates and using fundamental concepts from linear algebra, simple rules to calculate the first- and second-order derivatives of an internal coordinate can be formulated. To first order, they are not difficult to derive[64]. The purpose of this work is to find the corresponding simple rules to calculate the derivative of an internal coordinate to second order.

The choice of coordinate system to calculate the derivatives is important. As a molecular potential only depends on the internal coordinates, an obvious choice is to differentiate with respect to the internal coordinates. However, the internal coordinates most often used in normal mode analysis[61, 65, 66, 67] and geometry optimization[68] are curvilinear and redundant, and notoriously difficult to work with. In geometry optimization for example, the consequence of using redundant coordinates is that the Newton step has to be modified and that a conversion between internal and Cartesian coordinates to estimate the new Cartesian coordinates is necessary[69]. One of the advantages of calculating the derivatives of an internal coordinate in Cartesian coordinates is that the Cartesian coordinates are non-redundant. Hence, build-in optimization routines can be directly applied in geometry optimization.

Many methods exist to calculate the derivatives of an internal coordinate in Cartesian coordinates[64, 70, 71, 72, 73, 74]. The Wilson B-matrix[64] is a Jacobian matrix defined as the first derivatives of the internal coordinates with respect to the Cartesian coordinates. The Wilson B-matrix is required to calculate the gradient with respect to the Cartesian coordinates which is used in geometry optimization. Furthermore, its pseudo-inverse is important, for example to optimize molecular structures using internal coordinates instead of Cartesian coordinates[75, 63].

The first derivatives of an internal coordinate with respect to the Cartesian coordinates can be simplified if they are expressed in two orthonormal bases. Two orthonormal bases are connected by a transformation matrix which only depends on the bond angles and the torsion angles. In this approach, the bond angles and torsion angles act as Euler angles. It is well known that the B-matrix has a compact expression when using two orthonormal bases[64]. The configuration of a molecule can be generated from its internal coordinates using either Euler angles[76, 16] or geometric (Clifford) algebra[17]. The calculation of the first derivatives has therefore also been made using geometric algebra to avoid the Gimbal lock problem[77]. A method to calculate the second derivatives of an internal coordinate using a single orthonormal basis can be found in[78].

In this work, we express the block matrices that form the Hessian of an internal

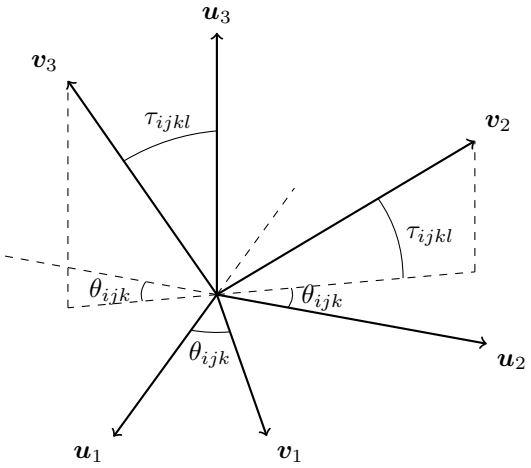


Figure 5.1: Showing how the coordinate system $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3\}$ can be calculated from $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3\}$ by first rotating the coordinate system about $\boldsymbol{u}_1$ using the torsion angle τ_{ijkl}, and thereafter rotating the coordinate system about $\boldsymbol{u}_3$ using the bond angle θ_{ijk}.

coordinate using two orthonormal bases. We show that each of the block matrices that form the Hessian can be expanded in terms of operators. This leads to simple rules for calculating the derivatives of an internal coordinate. It is useful to know how the Hessian is generated effectively. We therefore provide algorithms to calculate the Hessian of a function of an internal coordinate. Furthermore, we show how to implement the Hessian of a molecular potential in high-level programming languages based on vectorization. Finally, we use the results to derive a compact expression for the second-order expansion of a function of an internal coordinate.

5.2 Method of Calculation

5.2.1 Internal coordinates, Euler angles and orthonormal bases

We define $\boldsymbol{x}_i$, $\boldsymbol{x}_j$, $\boldsymbol{x}_k$ and $\boldsymbol{x}_l$ as the coordinates of the atoms i, j, k and l, respectively. For the bond length between the atoms i and j we use the notation

$$r_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|, \tag{5.1}$$

and similarly for r_{jk}, r_{kl} and r_{jl}. We define the unit vectors $\boldsymbol{u}_1$, $\boldsymbol{v}_1$ and $\boldsymbol{w}_1$ as the column matrices

$$\begin{aligned}
&\boldsymbol{u}_1 = \frac{1}{r_{ij}} \begin{bmatrix} x_i^{(1)} - x_j^{(1)} \\ x_i^{(2)} - x_j^{(2)} \\ x_i^{(3)} - x_j^{(3)} \end{bmatrix}, \ \boldsymbol{v}_1 = \frac{1}{r_{jk}} \begin{bmatrix} x_k^{(1)} - x_j^{(1)} \\ x_k^{(2)} - x_j^{(2)} \\ x_k^{(3)} - x_j^{(3)} \end{bmatrix} \text{ and } \boldsymbol{w}_1 = \frac{1}{r_{kl}} \begin{bmatrix} x_l^{(1)} - x_k^{(1)} \\ x_l^{(2)} - x_k^{(2)} \\ x_l^{(3)} - x_k^{(3)} \end{bmatrix} \\
&\text{(or } \boldsymbol{w}_1 = \frac{1}{r_{jl}} \begin{bmatrix} x_l^{(1)} - x_j^{(1)} \\ x_l^{(2)} - x_j^{(2)} \\ x_l^{(3)} - x_j^{(3)} \end{bmatrix} \text{)}.
\end{aligned} \tag{5.2}$$

Consider the two orthonormal bases $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3\}$ and $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3\}$ as shown in Figure 1. The bond angle θ_{ijk} and the torsion angle τ_{ijkl} are given by

$$\begin{aligned} \theta_{ijk} &= \cos^{-1}\left(\boldsymbol{u}_1^T \boldsymbol{v}_1\right) \\ \tau_{ijkl} &= \text{sign}\left(\boldsymbol{u}_3^T \boldsymbol{v}_2\right) \cos^{-1}\left(\boldsymbol{u}_3^T \boldsymbol{v}_3\right). \end{aligned} \tag{5.3}$$

We remark that a torsion angle is signed. The unit vector $\boldsymbol{u}_2$, orthogonal to $\boldsymbol{u}_1$, but in the plane spanned by $\boldsymbol{u}_1$ and $\boldsymbol{v}_1$ is given by the equation

$$\boldsymbol{u}_2 = \left(\boldsymbol{v}_1 - \cos\theta_{ijk}\boldsymbol{u}_1\right) / \sin\theta_{ijk}. \tag{5.4}$$

The transformation matrix mapping the coordinate system $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3\}$ to $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3\}$ is a product of two Euler rotations. A rotation about $\boldsymbol{u}_1$

$$\boldsymbol{R}_x(\tau_{ijkl}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\tau_{ijkl} & -\sin\tau_{ijkl} \\ 0 & \sin\tau_{ijkl} & \cos\tau_{ijkl} \end{bmatrix}, \tag{5.5}$$

followed by a rotation about $\boldsymbol{u}_3$

$$\boldsymbol{R}_z(\theta_{ijk}) = \begin{bmatrix} \cos\theta_{ijk} & -\sin\theta_{ijk} & 0 \\ \sin\theta_{ijk} & \cos\theta_{ijk} & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{5.6}$$

The product of the two rotation matrices is

$$\boldsymbol{S}(\theta_{ijk}, \tau_{ijkl}) = \begin{bmatrix} \cos\theta_{ijk} & -\sin\theta_{ijk}\cos\tau_{ijkl} & \sin\theta_{ijk}\sin\tau_{ijkl} \\ \sin\theta_{ijk} & \cos\theta_{ijk}\cos\tau_{ijkl} & -\cos\theta_{ijk}\sin\tau_{ijkl} \\ 0 & \sin\tau_{ijkl} & \cos\tau_{ijkl} \end{bmatrix}. \tag{5.7}$$

Given $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3\}$, we can find $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3\}$ using two rotations:

$$\boldsymbol{v}_i = \boldsymbol{S}(\theta_{ijk}, \tau_{ijkl})\boldsymbol{u}_i = \boldsymbol{R}_z(\theta_{ijk})\boldsymbol{R}_x(\tau_{ijkl})\boldsymbol{u}_i. \tag{5.8}$$

Similarly, when we consider the bases $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3\}$ and $\{\boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{w}_3\}$, the unit vector $\boldsymbol{v}_2$ can be written as

$$\boldsymbol{v}_2 = \left(\boldsymbol{w}_1 - \cos\theta_{jkl}\boldsymbol{v}_1\right) / \sin\theta_{jkl}, \tag{5.9}$$

where

$$\theta_{jkl} = \cos^{-1}\left(\boldsymbol{v}_1^T \boldsymbol{w}_1\right). \tag{5.10}$$

Thus, three Euler rotations are required to calculate $\{\boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{w}_3\}$ given $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3\}$:

$$\boldsymbol{w}_i = \boldsymbol{T}(\theta_{ijk}, \tau_{ijkl}, \theta_{jkl})\boldsymbol{u}_i = \boldsymbol{S}(\theta_{ijk}, \tau_{ijkl})\boldsymbol{R}_z(\theta_{jkl})\boldsymbol{u}_i = \boldsymbol{R}_z(\theta_{ijk})\boldsymbol{R}_x(\tau_{ijkl})\boldsymbol{R}_z(\theta_{jkl})\boldsymbol{u}_i, \tag{5.11}$$

where all of the matrices are fixed-axis rotations in the basis $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3\}$.

Finally, we note that a torsion angle is referred to as an improper torsion angle, when $\boldsymbol{w}_1$ is defined using the indices j and l instead of k and l in equation (5.2).

5.2.2 Motivation

To calculate the first and second derivatives of θ_{ijk} and τ_{ijkl}, we derive the derivatives of

$$F = \boldsymbol{u}_m^T \boldsymbol{v}_n, \tag{5.12}$$

In Cartesian coordinates, the first derivative of a function f with respect to atom α has the matrix representation

$$[\nabla_\alpha f] = \frac{\partial f}{\partial \boldsymbol{x}_\alpha} = \begin{bmatrix} \partial_{x_\alpha^{(1)}} f \\ \partial_{x_\alpha^{(2)}} f \\ \partial_{x_\alpha^{(3)}} f \end{bmatrix}, \tag{5.13}$$

and the derivative of $\frac{\partial f}{\partial \boldsymbol{x}_\alpha}$ with respect to atom β has the matrix representation

$$\left[\nabla^2_{\alpha,\beta} f\right] = \frac{\partial^2 f}{\partial \boldsymbol{x}_\alpha \partial \boldsymbol{x}_\beta} = \begin{bmatrix} \partial_{x_\alpha^{(1)}}\partial_{x_\beta^{(1)}} f & \partial_{x_\alpha^{(1)}}\partial_{x_\beta^{(2)}} f & \partial_{x_\alpha^{(1)}}\partial_{x_\beta^{(3)}} f \\ \partial_{x_\alpha^{(2)}}\partial_{x_\beta^{(1)}} f & \partial_{x_\alpha^{(2)}}\partial_{x_\beta^{(2)}} f & \partial_{x_\alpha^{(2)}}\partial_{x_\beta^{(3)}} f \\ \partial_{x_\alpha^{(3)}}\partial_{x_\beta^{(1)}} f & \partial_{x_\alpha^{(3)}}\partial_{x_\beta^{(2)}} f & \partial_{x_\alpha^{(3)}}\partial_{x_\beta^{(3)}} f \end{bmatrix}. \tag{5.14}$$

The first derivative of F with respect to atom α is given by

$$[\nabla_\alpha F] = \left(\frac{\partial \boldsymbol{u}_m}{\partial \boldsymbol{x}_\alpha}\right)^T \boldsymbol{v}_n + \left(\frac{\partial \boldsymbol{v}_n}{\partial \boldsymbol{x}_\alpha}\right)^T \boldsymbol{u}_m. \tag{5.15}$$

We remark that $[\nabla_\alpha F]$ is a column matrix. Thus, the last product is $\left(\frac{\partial \boldsymbol{v}_n}{\partial \boldsymbol{x}_\alpha}\right)^T \boldsymbol{u}_m$ and not $\boldsymbol{u}_m^T \left(\frac{\partial \boldsymbol{v}_n}{\partial \boldsymbol{x}_\alpha}\right)$.

We can express $[\nabla_\alpha F]$ in the orthonormal bases $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3\}$ and $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3\}$:

$$[\nabla_\alpha F] = \sum_{p=1}^{3} \left(a_p \boldsymbol{u}_p + b_p \boldsymbol{v}_p\right). \tag{5.16}$$

A derivation of the constants a_p and b_p for $[\nabla_\alpha r_{ij}]$, $[\nabla_\alpha \theta_{ijk}]$ and $[\nabla_\alpha \tau_{ijkl}]$ may be found in [64] or below. We could let a_p or b_p vanish but the most compact expressions are found when we use two bases rather than one.

The second derivative of F with respect to atom α and atom β is

$$\left[\nabla^2_{\alpha,\beta} F\right] = \sum_{p=1}^{3} \left(\frac{\partial \boldsymbol{u}_p}{\partial \boldsymbol{x}_\alpha} a_p + b_p \frac{\partial \boldsymbol{v}_p}{\partial \boldsymbol{x}_\alpha} + \boldsymbol{u}_p \left(\frac{\partial a_p}{\partial \boldsymbol{x}_\beta}\right)^T + \boldsymbol{v}_p \left(\frac{\partial b_p}{\partial \boldsymbol{x}_\beta}\right)^T \right). \tag{5.17}$$

The derivatives of a_p and $\boldsymbol{u}_p$ have a simple matrix representation in the basis $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3\}$ and the derivatives of b_p and $\boldsymbol{v}_p$ have a simple matrix representation in the basis $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3\}$. Thus, we can find a compact expression for $\left[\nabla^2_{\alpha,\beta} F\right]$ using the tensor product expansion[79]:

$$\left[\nabla^2_{\alpha,\beta} F\right] = \sum_{p=1}^{3}\sum_{q=1}^{3} \left(a_{pq} \boldsymbol{u}_p \boldsymbol{u}_q^T + b_{pq} \boldsymbol{v}_p \boldsymbol{v}_q^T\right). \tag{5.18}$$

The purpose of this work is to find a_{pq} and b_{pq} for $\left[\nabla^2_{\alpha,\beta} r_{ij}\right]$, $\left[\nabla^2_{\alpha,\beta} \theta_{ijk}\right]$ and $\left[\nabla^2_{\alpha,\beta} \tau_{ijkl}\right]$. Before we derive these results, we introduce a set of operators and derive the derivatives of the basis vectors. We do this in the following two sections.

5.2.3 Operators and their matrix representations

A tensor product expansion is an operator. It has a matrix representation which is defined by how it acts on basis vectors. An operator can be written as:

$$\boldsymbol{P} = \sum_{i=1}^{3}\sum_{j=1}^{3} \boldsymbol{u}_i \left(\boldsymbol{u}_i^T \boldsymbol{P} \boldsymbol{u}_j\right) \boldsymbol{u}_j^T. \tag{5.19}$$

The matrix representation of $\boldsymbol{P}_X$ defined by

$$\boldsymbol{P}_X = \boldsymbol{u}_1\boldsymbol{u}_2^T + \boldsymbol{u}_2\boldsymbol{u}_1^T, \tag{5.20}$$

is therefore

$$\boldsymbol{P}_X = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{5.21}$$

since its non-vanishing elements are

$$\begin{aligned} \boldsymbol{u}_1^T \boldsymbol{P}_X \boldsymbol{u}_2 &= 1 \\ \boldsymbol{u}_2^T \boldsymbol{P}_X \boldsymbol{u}_1 &= 1. \end{aligned} \tag{5.22}$$

Using the orthonormal basis $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3\}$ the operators used here are:

$$\begin{aligned}
\boldsymbol{P}_1 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{P}_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{P}_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
\boldsymbol{P}_{\perp_1} &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{P}_{\perp_2} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{P}_{\perp_3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\
\boldsymbol{P}_{X_1} &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad \boldsymbol{P}_{X_2} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{P}_{X_3} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\
\boldsymbol{P}_{+_1} &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{P}_{+_2} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{P}_{+_3} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\
\boldsymbol{P}_{-_1} &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad \boldsymbol{P}_{-_2} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{P}_{-_3} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.
\end{aligned} \tag{5.23}$$

The operators $\boldsymbol{Q}_1$, $\boldsymbol{Q}_2$, $\boldsymbol{Q}_3$ and so forth in the orthonormal basis $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3\}$ are defined in the same manner. Finally, we define $\boldsymbol{PQ}_i \equiv \boldsymbol{u}_i\boldsymbol{v}_i^T$ and $\boldsymbol{QP}_i \equiv \boldsymbol{v}_i\boldsymbol{u}_i^T$ for $i = 1, 2, 3$.

For the special case of $\tau_{ijkl} = 0$ where $\boldsymbol{S}(\theta_{ijk}, 0) = \boldsymbol{R}_z(\theta_{ijk})$ and

$$\boldsymbol{v}_2 = -\sin\theta_{ijk}\boldsymbol{u}_1 + \cos\theta_{ijk}\boldsymbol{u}_2, \tag{5.24}$$

the following abbreviations are used

$$\boldsymbol{P}_\perp = \boldsymbol{P}_{\perp_3},\ \boldsymbol{P}_X = \boldsymbol{P}_{X_3},\ \boldsymbol{P}_+ = \boldsymbol{P}_{+_3},\ \boldsymbol{P}_- = \boldsymbol{P}_{-_3}, \tag{5.25}$$

and similarly for $\boldsymbol{Q}_\perp$, $\boldsymbol{Q}_X$, $\boldsymbol{Q}_+$ and $\boldsymbol{Q}_-$.

An operator is defined by how it acts on basis vectors. It therefore depends on our choice of basis. If two operators $\boldsymbol{P}$ in $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3\}$ and $\boldsymbol{Q}$ in $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3\}$ represent the same transformation such as $\boldsymbol{P}_1$ and $\boldsymbol{Q}_1$ then they are related by the similarity transformation

$$\boldsymbol{Q} = \boldsymbol{S}(\theta_{ijk}, \tau_{ijkl})\boldsymbol{P}\boldsymbol{S}(\theta_{ijk}, \tau_{ijkl})^T, \tag{5.26}$$

for example

$$\boldsymbol{Q}_1 = \boldsymbol{S}(\theta_{ijk}, \tau_{ijkl}) \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \boldsymbol{S}(\theta_{ijk}, \tau_{ijkl})^T = \boldsymbol{v}_1 \boldsymbol{v}_1^T. \tag{5.27}$$

5.2.4 The derivatives of a basis vector

We have to derive the derivatives of the basis vectors in the orthonormal bases $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3\}$ and $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3\}$ to find the derivatives of an inner product of two basis vectors and thereby the derivatives of θ_{ijk} and τ_{ijkl}. Here we find the derivatives with respect to i, k and l. Because internal coordinates are translational invariant, we do not have to derive the derivatives with respect to j.

The derivatives of $\boldsymbol{u}_1$ with respect to i, k and l are calculated componentwise using equation (5.1). First, we derive the first-order derivative of r with respect to i. It is given by

$$[\nabla_i r_{ij}] = \begin{bmatrix} \frac{x_i^{(1)} - x_j^{(1)}}{r_{ij}} \\ \frac{x_i^{(2)} - x_j^{(2)}}{r_{ij}} \\ \frac{x_i^{(3)} - x_j^{(3)}}{r_{ij}} \end{bmatrix}, \tag{5.28}$$

so $[\nabla_i r_{ij}] = \boldsymbol{u}_1$ and it follows that $[\nabla_{i,i}^2 r_{ij}] = \frac{\partial \boldsymbol{u}_1}{\partial \boldsymbol{x}_i}$.

The second-order derivative of r_{ij} is

$$\left[\nabla_{i,i}^2 r_{ij}\right] = \begin{bmatrix} \frac{1}{r_{ij}}\left(1 - \left(\frac{x_i^{(1)} - x_j^{(1)}}{r_{ij}}\right)^2\right) & -\frac{\left(x_i^{(1)} - x_j^{(1)}\right)\left(x_i^{(2)} - x_j^{(2)}\right)}{r_{ij}^3} & -\frac{\left(x_i^{(1)} - x_j^{(1)}\right)\left(x_i^{(3)} - x_j^{(3)}\right)}{r_{ij}^3} \\ -\frac{\left(x_i^{(2)} - x_j^{(2)}\right)\left(x_i^{(1)} - x_j^{(1)}\right)}{r_{ij}^3} & \frac{1}{r_{ij}}\left(1 - \left(\frac{x_i^{(2)} - x_j^{(2)}}{r_{ij}}\right)^2\right) & -\frac{\left(x_i^{(2)} - x_j^{(2)}\right)\left(x_i^{(3)} - x_j^{(3)}\right)}{r_{ij}^3} \\ -\frac{\left(x_i^{(3)} - x_j^{(3)}\right)\left(x_i^{(1)} - x_j^{(1)}\right)}{r_{ij}^3} & -\frac{\left(x_i^{(3)} - x_j^{(3)}\right)\left(x_i^{(2)} - x_j^{(2)}\right)}{r_{ij}^3} & \frac{1}{r_{ij}}\left(1 - \left(\frac{x_i^{(3)} - x_j^{(3)}}{r_{ij}}\right)^2\right) \end{bmatrix}. \tag{5.29}$$

Because $\boldsymbol{P}_\perp = \boldsymbol{I} - \boldsymbol{u}_1 \boldsymbol{u}_1^T$ where $\boldsymbol{I}$ is the identity matrix, we have that the derivatives of $\boldsymbol{u}_1$ are given by

$$\begin{aligned} \frac{\partial \boldsymbol{u}_1}{\partial \boldsymbol{x}_i} &= \frac{1}{r_{ij}} \boldsymbol{P}_\perp \\ \frac{\partial \boldsymbol{u}_1}{\partial \boldsymbol{x}_k} &= \boldsymbol{0} \\ \frac{\partial \boldsymbol{u}_1}{\partial \boldsymbol{x}_l} &= \boldsymbol{0}. \end{aligned} \tag{5.30}$$

The derivatives of $\boldsymbol{v}_1$ are likewise

$$\begin{aligned}\frac{\partial \boldsymbol{v}_1}{\partial \boldsymbol{x}_i} &= \boldsymbol{0} \\ \frac{\partial \boldsymbol{v}_1}{\partial \boldsymbol{x}_k} &= \frac{1}{r_{ij}}\boldsymbol{Q}_\perp \\ \frac{\partial \boldsymbol{v}_1}{\partial \boldsymbol{x}_l} &= \boldsymbol{0}.\end{aligned} \tag{5.31}$$

Thus we see that the derivatives of $\boldsymbol{u}_1$ and $\boldsymbol{v}_1$ are given by the operators $\boldsymbol{P}_\perp$ and $\boldsymbol{Q}_\perp$, respectively. The derivatives of $\boldsymbol{u}_2$, $\boldsymbol{u}_3$ as well as $\boldsymbol{v}_2$ and $\boldsymbol{v}_3$ can also be expanded in terms of operators. These are found using the two equations above, the chain rule and the first derivatives of $\cos\theta_{ijk}$ which can be calculated as a product of an operator and a basis vector

$$\begin{aligned}\left[\nabla_i \cos\theta_{ijk}\right] &= \left(\frac{\partial \boldsymbol{u}_1}{\partial \boldsymbol{x}_i}\right)^T \boldsymbol{v}_1 = \frac{1}{r_{ij}}\boldsymbol{P}_\perp \boldsymbol{v}_1 = \frac{\sin\theta_{ijk}}{r_{ij}}\boldsymbol{u}_2 \\ \left[\nabla_k \cos\theta_{ijk}\right] &= \left(\frac{\partial \boldsymbol{v}_1}{\partial \boldsymbol{x}_k}\right)^T \boldsymbol{u}_1 = \frac{1}{r_{jk}}\boldsymbol{Q}_\perp \boldsymbol{u}_1 = -\frac{\sin\theta_{ijk}}{r_{jk}}\boldsymbol{v}_2,\end{aligned} \tag{5.32}$$

where $\boldsymbol{v}_2$ is defined by equation (5.24).

The derivatives of $\boldsymbol{u}_2$ with respect to i, k and l are:

$$\begin{aligned}\frac{\partial \boldsymbol{u}_2}{\partial \boldsymbol{x}_i} &= -\frac{1}{r_{ij}}\left\{\boldsymbol{P}_{+_3} + \frac{\cos\theta_{ijk}}{\sin\theta_{ijk}}\boldsymbol{P}_3\right\} \\ \frac{\partial \boldsymbol{u}_2}{\partial \boldsymbol{x}_k} &= \frac{1}{r_{jk}}\frac{1}{\sin\theta_{ijk}}\boldsymbol{P}_3 \\ \frac{\partial \boldsymbol{u}_2}{\partial \boldsymbol{x}_l} &= 0.\end{aligned} \tag{5.33}$$

As an example of how the calculations are made we derive the derivative of $\boldsymbol{u}_2$ defined by

$$\boldsymbol{u}_2 = \frac{1}{\sin\theta_{ijk}}\left\{\boldsymbol{v}_1 - \cos\theta_{ijk}\boldsymbol{u}_1\right\}, \tag{5.34}$$

with respect to i. It is

$$\begin{aligned}\frac{\partial \boldsymbol{u}_2}{\partial \boldsymbol{x}_i} &= \frac{1}{\sin\theta_{ijk}}\left\{-\boldsymbol{u}_2\left[\nabla_i \sin\theta_{ijk}\right]^T - \boldsymbol{u}_1\left[\nabla_i \cos\theta_{ijk}\right]^T - \cos\theta_{ijk}\frac{\partial \boldsymbol{u}_1}{\partial \boldsymbol{x}_i}\right\} \\ &= \frac{1}{\sin\theta_{ijk}}\left\{\frac{\cos\theta_{ijk}}{\sin\theta_{ijk}}\boldsymbol{u}_2\left[\nabla_i \cos\theta_{ijk}\right]^T - \boldsymbol{u}_1\left[\nabla_i \cos\theta_{ijk}\right]^T - \frac{\cos\theta_{ijk}}{r_{ij}}\boldsymbol{P}_\perp\right\} \\ &= \frac{1}{r_{ij}}\frac{1}{\sin\theta_{ijk}}\left\{\cos\theta_{ijk}\boldsymbol{u}_2\boldsymbol{u}_2^T - \sin\theta_{ijk}\boldsymbol{u}_1\boldsymbol{u}_2^T - \cos\theta_{ijk}\boldsymbol{P}_\perp\right\} \\ &= -\frac{1}{r_{ij}}\left\{\boldsymbol{P}_{+_3} + \frac{\cos\theta_{ijk}}{\sin\theta_{ijk}}\boldsymbol{P}_3\right\}.\end{aligned} \tag{5.35}$$

The derivatives of $\boldsymbol{u}_3 = \boldsymbol{u}_1 \times \boldsymbol{u}_2$ with respect to the atomic coordinates i, k and l are

$$\begin{aligned}\frac{\partial \boldsymbol{u}_3}{\partial \boldsymbol{x}_i} &= \frac{1}{r_{ij}}\left\{-\boldsymbol{P}_{+_2} + \frac{\cos\theta_{ijk}}{\sin\theta_{ijk}}\boldsymbol{P}_{+_1}\right\} \\ \frac{\partial \boldsymbol{u}_3}{\partial \boldsymbol{x}_k} &= -\frac{1}{r_{jk}}\frac{1}{\sin\theta_{ijk}}\boldsymbol{P}_{+_1} \\ \frac{\partial \boldsymbol{u}_3}{\partial \boldsymbol{x}_l} &= 0.\end{aligned} \tag{5.36}$$

For proper torsion angles where

$$\boldsymbol{v}_2 = \frac{1}{\sin\theta_{jkl}}\left\{\boldsymbol{w}_1 - \cos\theta_{jkl}\boldsymbol{v}_1\right\}, \tag{5.37}$$

equivalent results may be found by differentiating $\boldsymbol{v}_2$ and $\boldsymbol{v}_3$. These are

$$\begin{aligned}
\frac{\partial \boldsymbol{v}_2}{\partial \boldsymbol{x}_i} &= 0 \\
\frac{\partial \boldsymbol{v}_2}{\partial \boldsymbol{x}_k} &= -\frac{1}{r_{jk}}\left\{\boldsymbol{Q}_{+_3} + \frac{\cos\theta_{jkl}}{\sin\theta_{jkl}}\boldsymbol{Q}_3\right\} - \frac{1}{r_{kl}}\frac{1}{\sin\theta_{jkl}}\boldsymbol{Q}_3 \\
\frac{\partial \boldsymbol{v}_2}{\partial \boldsymbol{x}_l} &= \frac{1}{r_{kl}}\frac{1}{\sin\theta_{jkl}}\boldsymbol{Q}_3 \\
\frac{\partial \boldsymbol{v}_3}{\partial \boldsymbol{x}_i} &= 0 \\
\frac{\partial \boldsymbol{v}_3}{\partial \boldsymbol{x}_k} &= \frac{1}{r_{jk}}\left\{-\boldsymbol{Q}_{+_2} + \frac{\cos\theta_{jkl}}{\sin\theta_{jkl}}\boldsymbol{Q}_{+_1}\right\} + \frac{1}{r_{kl}}\frac{1}{\sin\theta_{jkl}}\boldsymbol{Q}_{+_1} \\
\frac{\partial \boldsymbol{v}_3}{\partial \boldsymbol{x}_l} &= -\frac{1}{r_{kl}}\frac{1}{\sin\theta_{jkl}}\boldsymbol{Q}_{+_1}.
\end{aligned} \tag{5.38}$$

Similarly, we find that the derivatives of $\boldsymbol{v_2}$ and $\boldsymbol{v_3}$ for improper angles are given by

$$\begin{aligned}
\frac{\partial \boldsymbol{v}_2}{\partial \boldsymbol{x}_i} &= 0 \\
\frac{\partial \boldsymbol{v}_2}{\partial \boldsymbol{x}_k} &= -\frac{1}{r_{jk}}\left\{\boldsymbol{Q}_{+_3} + \frac{\cos\theta_{jkl}}{\sin\theta_{jkl}}\boldsymbol{Q}_3\right\} \\
\frac{\partial \boldsymbol{v}_2}{\partial \boldsymbol{x}_l} &= \frac{1}{r_{jl}}\frac{1}{\sin\theta_{jkl}}\boldsymbol{Q}_3 \\
\frac{\partial \boldsymbol{v}_3}{\partial \boldsymbol{x}_i} &= 0 \\
\frac{\partial \boldsymbol{v}_3}{\partial \boldsymbol{x}_k} &= \frac{1}{r_{jk}}\left\{-\boldsymbol{Q}_{+_2} + \frac{\cos\theta_{jkl}}{\sin\theta_{jkl}}\boldsymbol{Q}_{+_1}\right\} \\
\frac{\partial \boldsymbol{v}_3}{\partial \boldsymbol{x}_l} &= -\frac{1}{r_{jl}}\frac{1}{\sin\theta_{jkl}}\boldsymbol{Q}_{+_1}.
\end{aligned} \tag{5.39}$$

5.3 The derivatives of an internal coordinate

The Hessian of an internal coordinate is derived by expanding its block matrices in terms of operators using two orthonormal bases, $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3\}$ and $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3\}$. The derivatives of both the proper and improper torsion angles are derived. The gradient of internal coordinates is derived as well for the sake of completeness, since it is required to calculate the Hessian of a function, f, dependent on a bond length, a bond angle or a torsion angle. We do not have to derive all the non-vanishing elements of the gradient and Hessian of f. This is due to the translational invariance of the internal coordinates, which means that

$$[\nabla_\alpha f] = -\sum_{\beta\neq\alpha}[\nabla_\beta f], \tag{5.40}$$

and

$$\left[\nabla^2_{\alpha,\alpha} f\right] = -\sum_{\alpha \neq \beta} \left[\nabla^2_{\alpha,\beta} f\right]. \tag{5.41}$$

Furthermore, the Hessian is symmetric:

$$\left[\nabla^2_{\alpha,\beta} f\right] = \left[\nabla^2_{\beta,\alpha} f\right]^T. \tag{5.42}$$

We therefore only have to calculate the gradient with respect to the atomic coordinates i for r_{ij}; i and k for θ_{ijk} as well as i, k and l for τ_{ijkl}. Similarly, to calculate the Hessian, one block matrix is required for r_{ij}

$$\left[\nabla^2_{i,i} f(r_{ij})\right], \tag{5.43}$$

three block matrices for θ_{ijk}

$$\left[\nabla^2_{i,i} f(\theta_{ijk})\right] \ , \left[\nabla^2_{k,k} f(\theta_{ijk})\right] \ , \left[\nabla^2_{i,k} f(\theta_{ijk})\right], \tag{5.44}$$

and six block matrices are required for τ_{ijkl}

$$\left[\nabla^2_{i,i} f(\tau_{ijkl})\right], \left[\nabla^2_{k,k} f(\tau_{ijkl})\right], \left[\nabla^2_{l,l} f(\tau_{ijkl})\right], \left[\nabla^2_{i,k} f(\tau_{ijkl})\right], \left[\nabla^2_{k,l} f(\tau_{ijkl})\right], \left[\nabla^2_{i,l} f(\tau_{ijkl})\right]. \tag{5.45}$$

Finally, the chain rule for the second-order derivative of a real-valued function of an internal coordinate is

$$\begin{aligned}
\left[\nabla^2_{\alpha,\beta} f(r_{ij})\right] &= \frac{\partial f}{\partial r_{ij}} \left[\nabla^2_{\alpha,\beta} r_{ij}\right] + \frac{\partial^2 f}{\partial r_{ij}^2} \left[\nabla_\alpha r_{ij}\right] \left[\nabla_\beta r_{ij}\right]^T \\
\left[\nabla^2_{\alpha,\beta} f(\theta_{ijk})\right] &= \frac{\partial f}{\partial \theta_{ijk}} \left[\nabla^2_{\alpha,\beta} \theta_{ijk}\right] + \frac{\partial^2 f}{\partial \theta_{ijk}^2} \left[\nabla_\alpha \theta_{ijk}\right] \left[\nabla_\beta \theta_{ijk}\right]^T \\
\left[\nabla^2_{\alpha,\beta} f(\tau_{ijkl})\right] &= \frac{\partial f}{\partial \tau_{ijkl}} \left[\nabla^2_{\alpha,\beta} \tau_{ijkl}\right] + \frac{\partial^2 f}{\partial \tau_{ijkl}^2} \left[\nabla_\alpha \tau_{ijkl}\right] \left[\nabla_\beta \tau_{ijkl}\right]^T.
\end{aligned} \tag{5.46}$$

5.3.1 The derivatives of a bond length

The derivatives of r_{ij} are calculated using equation (5.2). The first derivative is given by

$$\left[\nabla_i r_{ij}\right] = \boldsymbol{u}_1. \tag{5.47}$$

The second derivative of r_{ij} and the tensor product of the gradient vector above can be written as

$$\begin{aligned}
\left[\nabla^2_{i,i} r_{ij}\right] &= \frac{1}{r_{ij}} \boldsymbol{P}_\perp \\
\left[\nabla_i r_{ij}\right] \left[\nabla_i r_{ij}\right]^T &= \boldsymbol{P}_1,
\end{aligned} \tag{5.48}$$

hence the Hessian of a function $f(r_{ij})$ can be calculated using the algorithm:

1. Calculate r_{ij} and $\boldsymbol{u}_1$ using the equations (5.1) and (5.2).
2. Calculate $\boldsymbol{P}_1$ and $\boldsymbol{P}_\perp$.
3. Calculate $\frac{\partial f}{\partial r_{ij}}$ and $\frac{\partial^2 f}{\partial r_{ij}^2}$.
4. Calculate the block matrix $\left[\nabla^2_{i,i} f(r_{ij})\right]$ using the equations (5.46) and (5.48).
5. Calculate the remaining block matrices using the equations (5.41) and (5.42).

5.3.2 The derivatives of a bond angle

Next, the first-order derivatives of θ_{ijk} are given by

$$\begin{aligned}[\nabla_i \theta_{ijk}] &= -\frac{1}{r_{ij}} \boldsymbol{u}_2 \\ [\nabla_k \theta_{ijk}] &= \frac{1}{r_{jk}} \boldsymbol{v}_2.\end{aligned} \tag{5.49}$$

This result follows immediately by use of equation (5.32) and the chain rule.

The building blocks that form the Hessian of θ_{ijk} and functions of these are given by

$$\begin{aligned}\left[\nabla^2_{i,i} \theta_{ijk}\right] &= \frac{1}{r_{ij}^2} \frac{1}{\sin\theta_{ijk}} \left(\cos\theta_{ijk} \boldsymbol{P}_3 + \sin\theta_{ijk} \boldsymbol{P}_X\right) \\ \left[\nabla^2_{k,k} \theta_{ijk}\right] &= \frac{1}{r_{jk}^2} \frac{1}{\sin\theta_{ijk}} \left(\cos\theta_{ijk} \boldsymbol{P}_3 - \sin\theta_{ijk} \boldsymbol{Q}_X\right) \\ \left[\nabla^2_{i,k} \theta_{ijk}\right] &= -\frac{1}{r_{ij} r_{jk}} \frac{1}{\sin\theta_{ijk}} \boldsymbol{P}_3 \\ [\nabla_i \theta_{ijk}][\nabla_i \theta_{ijk}]^T &= \frac{1}{r_{ij}^2} \boldsymbol{P}_2 \\ [\nabla_k \theta_{ijk}][\nabla_k \theta_{ijk}]^T &= \frac{1}{r_{jk}^2} \boldsymbol{Q}_2 \\ [\nabla_i \theta_{ijk}][\nabla_k \theta_{ijk}]^T &= -\frac{1}{r_{ij} r_{jk}} \boldsymbol{P}\boldsymbol{Q}_2.\end{aligned} \tag{5.50}$$

As an example, we derive $[\nabla_{i,i} \theta_{ijk}]$. It is found by calculating the derivative of $[\nabla_i \theta_{ijk}]$ with respect to the i-th atom using equation (5.49)

$$\left[\nabla^2_{i,i} \theta_{ijk}\right] = \frac{1}{r_{ij}^2} \boldsymbol{u}_2 \boldsymbol{u}_1^T - \frac{1}{r_{ij}} \frac{\partial \boldsymbol{u}_2}{\partial \boldsymbol{x}_i}. \tag{5.51}$$

The derivatives of the basis vectors were derived above, and by inserting the expression for $\frac{\partial \boldsymbol{u}_2}{\partial \boldsymbol{x}_i}$ we get the result

$$\begin{aligned}\left[\nabla^2_{i,i} \theta_{ijk}\right] &= \frac{1}{r_{ij}^2} \boldsymbol{P}_- + \frac{1}{r_{ij}^2} \left\{ \boldsymbol{P}_+ + \frac{\cos\theta_{ijk}}{\sin\theta_{ijk}} \boldsymbol{P}_3 \right\} \\ &= \frac{1}{r_{ij}^2} \frac{1}{\sin\theta_{ijk}} \left(\cos\theta_{ijk} \boldsymbol{P}_3 + \sin\theta_{ijk} \boldsymbol{P}_X\right).\end{aligned} \tag{5.52}$$

The algorithm to calculate the Hessian of $f(\theta_{ijk})$ is:

1. Calculate r_{ij}, r_{jk}, $\cos\theta_{ijk}$, $\boldsymbol{u}_1$, $\boldsymbol{u}_2$, $\boldsymbol{u}_3$, $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ using the equations (5.1) - (5.4) and (5.24).
2. Calculate $\boldsymbol{P}_2$, $\boldsymbol{P}_X$, $\boldsymbol{P}_3$, $\boldsymbol{Q}_2$, $\boldsymbol{Q}_X$, and $\boldsymbol{P}\boldsymbol{Q}_2$.
3. Calculate $\frac{\partial f}{\partial \theta_{ijk}}$ and $\frac{\partial^2 f}{\partial \theta_{ijk}^2}$.
4. Calculate the block matrices $\left[\nabla^2_{i,i} f(\theta_{ijk})\right]$, $\left[\nabla^2_{k,k} f(\theta_{ijk})\right]$ and $\left[\nabla^2_{i,k} f(\theta_{ijk})\right]$ using the equations (5.46) and (5.50).
5. Calculate the remaining block matrices using the equations (5.41) and (5.42).

5.3.3 The derivatives of a torsion angle

First, we consider the proper torsion angles. The vectors that form the gradient of τ_{ijkl} are given by:

$$\begin{aligned}
[\nabla_i \tau_{ijkl}] &= \frac{1}{r_{ij}} \frac{1}{\sin\theta_{ijk}} \boldsymbol{u}_3 \\
[\nabla_k \tau_{ijkl}] &= -\frac{1}{r_{jk}} \left\{ \frac{\cos\theta_{ijk}}{\sin\theta_{ijk}} \boldsymbol{u}_3 + \frac{\cos\theta_{jkl}}{\sin\theta_{jkl}} \boldsymbol{v}_3 \right\} - \frac{1}{r_{kl}} \frac{1}{\sin\theta_{jkl}} \boldsymbol{v}_3 \\
[\nabla_l \tau_{ijkl}] &= \frac{1}{r_{kl}} \frac{1}{\sin\theta_{jkl}} \boldsymbol{v}_3.
\end{aligned} \tag{5.53}$$

Equivalent to the calculation of the gradient of θ_{ijk} above, we may derive the expressions using

$$\begin{aligned}
[\nabla_i \cos\tau_{ijkl}] &= \left(\frac{\partial \boldsymbol{u}_3}{\partial \boldsymbol{x}_i} \right)^T \boldsymbol{v}_3 \\
[\nabla_k \cos\tau_{ijkl}] &= \left(\frac{\partial \boldsymbol{u}_3}{\partial \boldsymbol{x}_k} \right)^T \boldsymbol{v}_3 + \left(\frac{\partial \boldsymbol{v}_3}{\partial \boldsymbol{x}_k} \right)^T \boldsymbol{u}_3 \\
[\nabla_l \cos\tau_{ijkl}] &= \left(\frac{\partial \boldsymbol{v}_3}{\partial \boldsymbol{x}_l} \right)^T \boldsymbol{u}_3.
\end{aligned} \tag{5.54}$$

The three formulas are obtained by evaluating the products of the operators and the basis vectors. The second derivatives of the proper torsion angles are calculated using the analytical expressions for $[\nabla_i \tau_{ijkl}]$, $[\nabla_k \tau_{ijkl}]$ and $[\nabla_l \tau_{ijkl}]$. The derivative of $[\nabla_l \tau_{ijkl}]$ with respect to the l-th atom, for example, is given by

$$\left[\nabla^2_{l,l} \tau_{ijkl}\right] = -\frac{1}{r^2_{kl}} \frac{1}{\sin\theta_{jkl}} \boldsymbol{v}_3 \boldsymbol{w}_1^T - \frac{1}{r_{kl}} \frac{1}{\sin^2\theta_{jkl}} \boldsymbol{v}_3 \left[\nabla_l \sin\theta_{jkl}\right]^T + \frac{1}{r_{kl}} \frac{1}{\sin\theta_{jkl}} \frac{\partial \boldsymbol{v}_3}{\partial \boldsymbol{x}_l}. \tag{5.55}$$

We can also write this as

$$\begin{aligned}
\left[\nabla^2_{l,l} \tau_{ijkl}\right] = &\frac{1}{r^2_{kl}} \frac{1}{\sin^2\theta_{jkl}} \left\{ -\sin\theta_{jkl} \cos\theta_{jkl} \boldsymbol{v}_3 \boldsymbol{v}_1^T - \sin^2\theta_{jkl} \boldsymbol{v}_3 \boldsymbol{v}_2^T \right\} \\
&+ \frac{1}{r_{kl}} \frac{1}{\sin^2\theta_{jkl}} \left\{ \frac{\cos\theta_{jkl}}{r_{kl}} \left(\sin\theta_{jkl} \boldsymbol{v}_3 \boldsymbol{v}_1^T - \cos\theta_{jkl} \boldsymbol{v}_3 \boldsymbol{v}_2^T \right) \right\} \\
&- \frac{1}{r^2_{kl}} \frac{1}{\sin^2\theta_{jkl}} \boldsymbol{v}_2 \boldsymbol{v}_3^T.
\end{aligned} \tag{5.56}$$

Now, two terms cancel and two terms can be simplified using the Pythagorean identity. It follows that

$$\begin{aligned}
\left[\nabla^2_{l,l} \tau_{ijkl}\right] &= -\frac{1}{r^2_{kl}} \frac{1}{\sin^2\theta_{jkl}} \left\{ \boldsymbol{v}_2 \boldsymbol{v}_3^T + \boldsymbol{v}_3 \boldsymbol{v}_2^T \right\} \\
&= -\frac{1}{r^2_{kl}} \frac{1}{\sin^2\theta_{jkl}} \boldsymbol{Q}_{X_1}.
\end{aligned} \tag{5.57}$$

The other block matrices may be calculated in an equivalent manner. The individual block matrices are given by

$$
\begin{aligned}
\left[\nabla^2_{i,i}\tau_{ijkl}\right] &= \frac{1}{r_{ij}^2}\frac{1}{\sin^2\theta_{ijk}}\left\{\cos\theta_{ijk}\boldsymbol{P}_{X_1} - \sin\theta_{ijk}\boldsymbol{P}_{X_2}\right\} \\
\left[\nabla^2_{k,k}\tau_{ijkl}\right] &= \frac{1}{r_{jk}^2}\left\{\frac{\cos\theta_{ijk}}{\sin^2\theta_{ijk}}\boldsymbol{P}_{X_1} + \cos\theta_{ijk}\boldsymbol{P}_{-_1} - \sin\theta_{ijk}\boldsymbol{P}_{-_2} - \boldsymbol{Q}_{-_1}\right\} \\
&\quad - \frac{1}{\sin^2\theta_{jkl}}\left(\frac{1}{r_{kl}} + \frac{1}{r_{jk}}\cos\theta_{jkl}\right)^2\boldsymbol{Q}_{X_1} + \frac{1}{\sin\theta_{jkl}}\left(\frac{1}{r_{jk}r_{kl}} + \frac{1}{r_{jk}^2}\cos\theta_{jkl}\right)\boldsymbol{Q}_{X_2} \\
\left[\nabla^2_{l,l}\tau_{ijkl}\right] &= -\frac{1}{r_{kl}^2}\frac{1}{\sin^2\theta_{jkl}}\boldsymbol{Q}_{X_1} \\
\left[\nabla^2_{i,k}\tau_{ijkl}\right] &= \frac{1}{r_{ij}r_{jk}}\frac{1}{\sin^2\theta_{ijk}}\left\{-\boldsymbol{P}_{X_1} + \cos\theta_{ijk}\sin\theta_{ijk}\boldsymbol{P}_{-_2} + \sin^2\theta_{ijk}\boldsymbol{P}_{-_1}\right\} \\
\left[\nabla^2_{k,l}\tau_{ijkl}\right] &= \frac{1}{\sin^2\theta_{jkl}}\left(\frac{1}{r_{kl}^2} + \frac{1}{r_{jk}r_{kl}}\cos\theta_{jkl}\right)\boldsymbol{Q}_{X_1} - \frac{1}{r_{jk}r_{kl}}\frac{1}{\sin\theta_{jkl}}\boldsymbol{Q}_{-_2} \\
\left[\nabla^2_{i,l}\tau_{ijkl}\right] &= 0.
\end{aligned}
\tag{5.58}
$$

We note that the block matrices $\left[\nabla^2_{i,i}\tau_{ijkl}\right]$, $\left[\nabla^2_{k,k}\tau_{ijkl}\right]$ and $\left[\nabla^2_{l,l}\tau_{ijkl}\right]$ are symmetric. The tensor products of the vectors that form the gradient of proper torsion angles are necessary to calculate the Hessian of functions of torsion angles. These tensor products of the gradient vectors are

$$
\begin{aligned}
\left[\nabla_i\tau_{ijkl}\right]\left[\nabla_i\tau_{ijkl}\right]^T &= \frac{1}{r_{ij}^2}\frac{1}{\sin^2\theta_{ijk}}\boldsymbol{P}_3 \\
\left[\nabla_k\tau_{ijkl}\right]\left[\nabla_k\tau_{ijkl}\right]^T &= \frac{1}{r_{jk}^2}\frac{\cos^2\theta_{ijk}}{\sin^2\theta_{ijk}}\boldsymbol{P}_3 + \left(\frac{1}{r_{kl}}\frac{1}{\sin\theta_{jkl}} + \frac{1}{r_{jk}}\frac{\cos\theta_{jkl}}{\sin\theta_{jkl}}\right)^2\boldsymbol{Q}_3 \\
&\quad + \frac{\cos\theta_{ijk}}{\sin\theta_{ijk}\sin\theta_{jkl}}\left(\frac{1}{r_{jk}r_{kl}} + \frac{1}{r_{jk}^2}\cos\theta_{jkl}\right)\left\{\boldsymbol{P}\boldsymbol{Q}_3 + \boldsymbol{Q}\boldsymbol{P}_3\right\} \\
\left[\nabla_l\tau_{ijkl}\right]\left[\nabla_l\tau_{ijkl}\right]^T &= \frac{1}{r_{kl}^2}\frac{1}{\sin^2\theta_{jkl}}\boldsymbol{Q}_3 \\
\left[\nabla_i\tau_{ijkl}\right]\left[\nabla_k\tau_{ijkl}\right]^T &= -\frac{1}{r_{ij}r_{jk}}\frac{\cos\theta_{ijk}}{\sin^2\theta_{ijk}}\boldsymbol{P}_3 - \frac{1}{r_{ij}r_{jk}}\frac{\cos\theta_{jkl}}{\sin\theta_{ijk}\sin\theta_{jkl}}\boldsymbol{P}\boldsymbol{Q}_3 - \frac{1}{r_{ij}r_{kl}}\frac{1}{\sin\theta_{ijk}\sin\theta_{jkl}}\boldsymbol{P}\boldsymbol{Q}_3 \\
\left[\nabla_k\tau_{ijkl}\right]\left[\nabla_l\tau_{ijkl}\right]^T &= -\frac{1}{\sin^2\theta_{jkl}}\left(\frac{1}{r_{kl}^2} + \frac{1}{r_{jk}r_{kl}}\cos\theta_{jkl}\right)\boldsymbol{Q}_3 - \frac{1}{r_{jk}r_{kl}}\frac{\cos\theta_{ijk}}{\sin\theta_{ijk}\sin\theta_{jkl}}\boldsymbol{P}\boldsymbol{Q}_3 \\
\left[\nabla_i\tau_{ijkl}\right]\left[\nabla_l\tau_{ijkl}\right]^T &= \frac{1}{r_{ij}r_{kl}}\frac{1}{\sin\theta_{ijk}\sin\theta_{jkl}}\boldsymbol{P}\boldsymbol{Q}_3.
\end{aligned}
\tag{5.59}
$$

The derivation of the derivatives of improper torsion angles is similar to the derivation of the derivatives of proper torsion angles. The difference is that $\boldsymbol{w}_1$ now depends on j and l instead of k and l (cf. equation (5.2)). This leads to different expressions for

the derivatives of τ_{ijkl}. The first derivatives of τ_{ijkl} are

$$\begin{aligned}
\left[\nabla_i \tau_{ijkl}\right] &= \frac{1}{r_{ij}} \frac{1}{\sin\theta_{ijk}} \boldsymbol{u}_3 \\
\left[\nabla_k \tau_{ijkl}\right] &= -\frac{1}{r_{jk}} \left\{ \frac{\cos\theta_{ijk}}{\sin\theta_{ijk}} \boldsymbol{u}_3 + \frac{\cos\theta_{jkl}}{\sin\theta_{jkl}} \boldsymbol{v}_3 \right\} \\
\left[\nabla_l \tau_{ijkl}\right] &= \frac{1}{r_{jl}} \frac{1}{\sin\theta_{jkl}} \boldsymbol{v}_3,
\end{aligned} \tag{5.60}$$

while the second derivatives are

$$\begin{aligned}
\left[\nabla^2_{i,i} \tau_{ijkl}\right] &= \frac{1}{r^2_{ij}} \frac{1}{\sin^2\theta_{ijk}} \left\{ \cos\theta_{ijk} \boldsymbol{P}_{X_1} - \sin\theta_{ijk} \boldsymbol{P}_{X_2} \right\} \\
\left[\nabla^2_{k,k} \tau_{ijkl}\right] &= \frac{1}{r^2_{jk}} \left\{ \frac{\cos\theta_{ijk}}{\sin^2\theta_{ijk}} \boldsymbol{P}_{X_1} + \cos\theta_{ijk} \boldsymbol{P}_{-_1} - \sin\theta_{ijk} \boldsymbol{P}_{-_2} - \boldsymbol{Q}_{-_1} - \frac{\cos^2\theta_{jkl}}{\sin^2\theta_{jkl}} \boldsymbol{Q}_{X_1} + \frac{\cos\theta_{jkl}}{\sin\theta_{jkl}} \boldsymbol{Q}_{X_2} \right\} \\
\left[\nabla^2_{l,l} \tau_{ijkl}\right] &= -\frac{1}{r^2_{jl}} \frac{1}{\sin^2\theta_{jkl}} \boldsymbol{Q}_{X_1} \\
\left[\nabla^2_{i,k} \tau_{ijkl}\right] &= \frac{1}{r_{ij} r_{jk}} \frac{1}{\sin^2\theta_{ijk}} \left\{ -\boldsymbol{P}_{X_1} + \sin^2\theta_{ijk} \boldsymbol{P}_{-_1} + \cos\theta_{ijk} \sin\theta_{ijk} \boldsymbol{P}_{-_2} \right\} \\
\left[\nabla^2_{k,l} \tau_{ijkl}\right] &= \frac{1}{r_{jk} r_{jl}} \frac{1}{\sin^2\theta_{jkl}} \left\{ \cos\theta_{jkl} \boldsymbol{Q}_{X_1} - \sin\theta_{jkl} \boldsymbol{Q}_{-_2} \right\} \\
\left[\nabla^2_{i,l} \tau_{ijkl}\right] &= 0.
\end{aligned} \tag{5.61}$$

The tensor products of the gradient vectors can easily be derived. They are

$$\begin{aligned}
\left[\nabla_i \tau_{ijkl}\right] \left[\nabla_i \tau_{ijkl}\right]^T &= \frac{1}{r^2_{ij}} \frac{1}{\sin^2\theta_{ijk}} \boldsymbol{P}_3 \\
\left[\nabla_k \tau_{ijkl}\right] \left[\nabla_k \tau_{ijkl}\right]^T &= \frac{1}{r^2_{jk}} \left\{ \frac{\cos^2\theta_{ijk}}{\sin^2\theta_{ijk}} \boldsymbol{P}_3 + \frac{\cos^2\theta_{jkl}}{\sin^2\theta_{jkl}} \boldsymbol{Q}_3 + \frac{\cos\theta_{ijk}}{\sin\theta_{ijk}} \frac{\cos\theta_{jkl}}{\sin\theta_{jkl}} \left\{ \boldsymbol{P}\boldsymbol{Q}_3 + \boldsymbol{Q}\boldsymbol{P}_3 \right\} \right\} \\
\left[\nabla_l \tau_{ijkl}\right] \left[\nabla_l \tau_{ijkl}\right]^T &= \frac{1}{r^2_{jl}} \frac{1}{\sin^2\theta_{jkl}} \boldsymbol{Q}_3 \\
\left[\nabla_i \tau_{ijkl}\right] \left[\nabla_k \tau_{ijkl}\right]^T &= -\frac{1}{r_{ij} r_{jk}} \frac{1}{\sin^2\theta_{ijk}} \left\{ \cos\theta_{ijk} \boldsymbol{P}_3 + \sin\theta_{ijk} \frac{\cos\theta_{jkl}}{\sin\theta_{jkl}} \boldsymbol{P}\boldsymbol{Q}_3 \right\} \\
\left[\nabla_k \tau_{ijkl}\right] \left[\nabla_l \tau_{ijkl}\right]^T &= -\frac{1}{r_{jk} r_{jl}} \frac{1}{\sin^2\theta_{jkl}} \left\{ \cos\theta_{jkl} \boldsymbol{Q}_3 + \sin\theta_{jkl} \frac{\cos\theta_{ijk}}{\sin\theta_{ijk}} \boldsymbol{P}\boldsymbol{Q}_3 \right\} \\
\left[\nabla_i \tau_{ijkl}\right] \left[\nabla_l \tau_{ijkl}\right]^T &= \frac{1}{r_{ij} r_{jl}} \frac{1}{\sin\theta_{ijk} \sin\theta_{jkl}} \boldsymbol{P}\boldsymbol{Q}_3.
\end{aligned} \tag{5.62}$$

The algorithm to generate the Hessian of $f(\tau_{ijkl})$ proceeds as follows:

1. Calculate r_{ij}, r_{jk}, r_{kl} (or r_{jl} for improper torsion angles), $\cos\theta_{ijk}$, $\sin\theta_{ijk}$, $\cos\theta_{jkl}$, $\sin\theta_{jkl}$, $\boldsymbol{u}_1$, $\boldsymbol{u}_2$, $\boldsymbol{u}_3$, $\boldsymbol{v}_1$, $\boldsymbol{v}_2$ and $\boldsymbol{v}_3$ using the equations (5.1) - (5.4) and (5.9) - (5.10).
2. Calculate $\boldsymbol{P}_{-_1}$, $\boldsymbol{P}_{-_2}$, $\boldsymbol{P}_{X_1}$, $\boldsymbol{P}_{X_2}$, $\boldsymbol{P}_3$, $\boldsymbol{Q}_{-_1}$, $\boldsymbol{Q}_{-_2}$, $\boldsymbol{Q}_{X_1}$, $\boldsymbol{Q}_{X_2}$, $\boldsymbol{Q}_3$, $\boldsymbol{P}\boldsymbol{Q}_3$ and $\boldsymbol{Q}\boldsymbol{P}_3$.
3. Calculate $\frac{\partial f}{\partial \tau_{ijkl}}$ and $\frac{\partial^2 f}{\partial \tau^2_{ijkl}}$.
4. Calculate the block matrices $\left[\nabla^2_{i,i} f(\tau_{ijkl})\right]$, $\left[\nabla^2_{k,k} f(\tau_{ijkl})\right]$, $\left[\nabla^2_{l,l} f(\tau_{ijkl})\right]$, $\left[\nabla^2_{i,k} f(\tau_{ijkl})\right]$, $\left[\nabla^2_{k,l} f(\tau_{ijkl})\right]$ and $\left[\nabla^2_{i,l} f(\tau_{ijkl})\right]$ using the equations (5.46) and (5.58) - (5.59) for proper torsion angles or (5.61) - (5.62) for improper torsion angles.

5. Calculate the remaining block matrices using the equations (5.41) and (5.42).

Finally, we remark that the formulas for the derivatives of τ_{ijkl} do not explicitly depend on τ_{ijkl}. They only have singularities when the bond lengths and bond angles vanish and thus no singularities at $\tau_{ijkl} = 0$ unless $\frac{\partial f}{\partial \tau_{ijkl}}$ or $\frac{\partial^2 f}{\partial \tau_{ijkl}^2}$ are singular. Under the assumption that the function is well-defined, we can therefore generate the gradient and Hessian of an arbitrary functional form dependent on τ_{ijkl} .

5.4 Applications

5.4.1 The Hessian of a molecular potential

The results in the preceding section constitute the necessary building blocks to generate the Hessian of a molecular potential only dependent on r_{ij}, θ_{ijk} and τ_{ijkl}. This is used for example in geometry optimization and molecular dynamics. An example of a molecular potential is the ENCAD potential[80]:

$$\begin{aligned}\text{ENCAD} = &\sum \frac{1}{2} K_{r_{ij}} \left(r_{ij} - r_0^{ij}\right)^2 + \sum \frac{1}{2} K_{\theta_{ijk}} \left(\theta_{ijk} - \theta_0^{ijk}\right)^2 + \sum \frac{1}{2} K_{\tau_{ijkl}} \left[1 - \cos(n\tau_{ijkl} + \tau_0^{ijkl})\right] \\ &+ \sum \epsilon_{r_{ij}} \left[(r_0^{ij}/r_{ij})^{12} - 2(r_0^{ij}/r_{ij})^6\right] + \sum (q_i q_j / r_{ij}).\end{aligned} \tag{5.63}$$

It consists of a sum of harmonic potentials about equilibrium bond lengths and bond angles, Fourier terms about equilibrium proper torsion angles where n expresses the periodicity of the rotational barrier, and Van der Waals and electrostatics terms which only depend on non-local interatomic distances.

In high-level programming languages such as Matlab or Octave, the efficiency of the procedure generating the Hessian of a molecular potential is significantly decreased when using loops. One of the advantages of the algorithms presented here is that vectorization techniques can be used. This is done by calculating the unit vectors and tensor products by vectorizing loops, indexing and the use of fast built-in functions such as `accumarray`. We thereby achieve high efficiency and accuracy in these programming languages.

5.4.2 Second-order expansions

The formulas for the derivatives of internal coordinates are used to find formulas for a second order expansion of a function of internal coordinates. The method is to write the second order expansion in its matrix form and to express the variables in the bases $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3\}$ and $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3\}$. This leads to compact expressions since each of the operators have a simple matrix representation in one of these bases.

For a molecule with coordinates $\boldsymbol{x}$, the second order expansion of the composite

function $f(g(\boldsymbol{x}))$ of the real-valued functions f and g in the direction of $\Delta\boldsymbol{x}$ is given by

$$\begin{aligned} f(g(\boldsymbol{x}+\Delta\boldsymbol{x})) &\approx f(g(\boldsymbol{x})) + \frac{\partial f}{\partial g}\sum_{\alpha}\Delta\boldsymbol{x}_{\alpha}^{T}\left[\nabla_{\alpha}g\right] + \frac{1}{2}\frac{\partial f}{\partial g}\sum_{\alpha,\beta}\Delta\boldsymbol{x}_{\alpha}^{T}\left[\nabla_{\alpha,\beta}^{2}g\right]\Delta\boldsymbol{x}_{\beta} \\ &\quad + \frac{1}{2}\frac{\partial^{2} f}{\partial g^{2}}\sum_{\alpha,\beta}\Delta\boldsymbol{x}_{\alpha}^{T}\left[\nabla_{\alpha}g\right]\left[\nabla_{\beta}g\right]^{T}\Delta\boldsymbol{x}_{\beta} \\ &= f(g) + \frac{\partial f}{\partial g}h_{1,g} + \frac{1}{2}\frac{\partial f}{\partial g}h_{2,g} + \frac{1}{2}\frac{\partial^{2} f}{\partial g^{2}}h_{1,g}^{2}, \end{aligned} \tag{5.64}$$

where $h_{1,g}=\sum_{\alpha}\Delta\boldsymbol{x}_{\alpha}^{T}\left[\nabla_{\alpha}g\right]$ and $h_{2,g}=\sum_{\alpha,\beta}\Delta\boldsymbol{x}_{\alpha}^{T}\left[\nabla_{\alpha,\beta}^{2}g\right]\Delta\boldsymbol{x}_{\beta}$. We consider the second order expansion of a function dependent on one or several internal coordinates. For $f(r_{ij},\theta_{ijk})$, for example, it is given by

$$\begin{aligned} f(r_{ij}(\boldsymbol{x}+\Delta\boldsymbol{x}),\theta_{ijk}(\boldsymbol{x}+\Delta\boldsymbol{x})) \approx & f(r_{ij},\theta_{ijk}) + \frac{\partial f}{\partial r_{ij}}h_{1,r_{ij}} + \frac{\partial f}{\partial\theta_{ijk}}h_{1,\theta_{ijk}} + \frac{1}{2}\frac{\partial f}{\partial r_{ij}}h_{2,r_{ij}} + \frac{1}{2}\frac{\partial f}{\partial\theta_{ijk}}h_{2,\theta_{ijk}} \\ & + \frac{1}{2}\frac{\partial^{2} f}{\partial r_{ij}^{2}}h_{1,r_{ij}}^{2} + \frac{1}{2}\frac{\partial^{2} f}{\partial\theta_{ijk}^{2}}h_{1,\theta_{ijk}}^{2} + \frac{\partial^{2} f}{\partial r_{ij}\theta_{ijk}}h_{1,r_{ij}}h_{1,\theta_{ijk}} \end{aligned} \tag{5.65}$$

Hence, it is sufficient only to calculate the terms $h_{1r_{ij}}$, $h_{1,\theta_{ijk}}$, $h_{1,\tau_{ijkl}}$, $h_{2,r_{ij}}$, $h_{2,\theta_{ijk}}$ and $h_{2,\tau_{ijkl}}$. In matrix form, these are:

$$\begin{aligned} h_{1,r_{ij}} &= \boldsymbol{a}^{T}\boldsymbol{u}_{1},\ h_{1,\theta_{ijk}} = \begin{bmatrix}\boldsymbol{a}^{T} & \boldsymbol{c}^{T}\end{bmatrix}\begin{bmatrix}-\frac{1}{r_{ij}}\boldsymbol{u}_{2}\\ \frac{1}{r_{jk}}\boldsymbol{v}_{2}\end{bmatrix},\ h_{1,\tau_{ijkl}} = \begin{bmatrix}\boldsymbol{a}^{T} & \boldsymbol{b}^{T} & \boldsymbol{c}^{T} & \boldsymbol{d}^{T}\end{bmatrix}\begin{bmatrix}\frac{1}{r_{ij}}\frac{1}{\sin\theta_{ijk}}\boldsymbol{u}_{3}\\ -\frac{1}{r_{jk}}\frac{\cos\theta_{ijk}}{\sin\theta_{ijk}}\boldsymbol{u}_{3}\\ -\frac{1}{r_{jk}}\frac{\cos\theta_{jkl}}{\sin\theta_{jkl}}\boldsymbol{v}_{3}\\ \frac{1}{r_{kl}}\frac{1}{\sin\theta_{jkl}}\boldsymbol{v}_{3}\end{bmatrix} \\ h_{2,r_{ij}} &= \boldsymbol{a}^{T}\left[\frac{1}{r_{ij}}\boldsymbol{P}_{\perp}\right]\boldsymbol{a},\ h_{2,\theta_{ijk}} = \begin{bmatrix}\boldsymbol{a}^{T} & \boldsymbol{b}^{T}\end{bmatrix}\begin{bmatrix}\frac{1}{r_{ij}^{2}}\left\{\frac{\cos\theta_{ijk}}{\sin\theta_{ijk}}\boldsymbol{P}_{3}+\boldsymbol{P}_{X}\right\} & -\frac{1}{r_{ij}r_{jk}}\frac{1}{\sin\theta_{ijk}}\boldsymbol{P}_{3}\\ -\frac{1}{r_{ij}r_{jk}}\frac{1}{\sin\theta_{ijk}}\boldsymbol{P}_{3} & \frac{1}{r_{jk}^{2}}\frac{\cos\theta_{ijk}}{\sin\theta_{ijk}}\boldsymbol{P}_{3}\end{bmatrix}\begin{bmatrix}\boldsymbol{a}\\ \boldsymbol{b}\end{bmatrix} + \boldsymbol{c}^{T}\left[-\frac{1}{r_{jk}^{2}}\boldsymbol{Q}_{X}\right]\boldsymbol{c} \\ h_{2,\tau_{ijkl}} &= \begin{bmatrix}\boldsymbol{a}^{T} & \boldsymbol{b}^{T}\end{bmatrix}\begin{bmatrix}\frac{1}{r_{ij}^{2}}\frac{1}{\sin^{2}\theta_{ijk}}\left\{\cos\theta_{ijk}\boldsymbol{P}_{X_1}-\sin\theta_{ijk}\boldsymbol{P}_{X_2}\right\} & \frac{1}{r_{ij}r_{jk}}\frac{1}{\sin^{2}\theta_{ijk}}\left\{-\boldsymbol{P}_{X_1}+\sin^{2}\theta_{ijk}\boldsymbol{P}_{-1}+\cos\theta_{ijk}\sin\theta_{ijk}\boldsymbol{P}_{-2}\right\}\\ \frac{1}{r_{ij}r_{jk}}\frac{1}{\sin^{2}\theta_{ijk}}\left\{-\boldsymbol{P}_{X_1}+\sin^{2}\theta_{ijk}\boldsymbol{P}_{+1}+\cos\theta_{ijk}\sin\theta_{ijk}\boldsymbol{P}_{+2}\right\} & \frac{1}{r_{jk}^{2}}\frac{1}{\sin^{2}\theta_{ijk}}\left\{\cos\theta_{ijk}\boldsymbol{P}_{X_1}+\cos\theta_{ijk}\sin^{2}\theta_{ijk}\boldsymbol{P}_{-1}-\sin^{3}\theta_{ijk}\boldsymbol{P}_{-2}\right\}\end{bmatrix} \\ &\quad + \begin{bmatrix}\boldsymbol{c}^{T} & \boldsymbol{d}^{T}\end{bmatrix}\begin{bmatrix}\frac{1}{r_{jk}^{2}}\frac{1}{\sin^{2}\theta_{jkl}}\left\{-\sin^{2}\theta_{jkl}\boldsymbol{Q}_{-1}-\cos^{2}\theta_{jkl}\boldsymbol{Q}_{X_1}+\cos\theta_{jkl}\sin\theta_{jkl}\boldsymbol{Q}_{X_2}\right\} & \frac{1}{r_{jk}r_{kl}}\frac{1}{\sin^{2}\theta_{jkl}}\left\{\cos\theta_{jkl}\boldsymbol{Q}_{X_1}-\sin\theta_{jkl}\boldsymbol{Q}_{-2}\right\}\\ \frac{1}{r_{jk}r_{kl}}\frac{1}{\sin^{2}\theta_{jkl}}\left\{\cos\theta_{jkl}\boldsymbol{Q}_{X_1}-\sin\theta_{jkl}\boldsymbol{Q}_{+2}\right\} & -\frac{1}{r_{kl}^{2}}\frac{1}{\sin^{2}\theta_{jkl}}\boldsymbol{Q}_{X_1}\end{bmatrix}\begin{bmatrix}\boldsymbol{c}\\ \boldsymbol{d}\end{bmatrix}, \end{aligned} \tag{5.66}$$

where we have used the notation

$$\begin{aligned} \boldsymbol{a} &= \Delta\boldsymbol{x}_{i} - \Delta\boldsymbol{x}_{j} \\ \boldsymbol{b} &= \Delta\boldsymbol{x}_{k} - \Delta\boldsymbol{x}_{j} \\ \boldsymbol{c} &= \boldsymbol{b} \\ \boldsymbol{d} &= \begin{cases}\Delta\boldsymbol{x}_{l} - \Delta\boldsymbol{x}_{k} \text{ for proper torsion angles}\\ \Delta\boldsymbol{x}_{l} - \Delta\boldsymbol{x}_{j} \text{ for improper torsion angles}\end{cases}. \end{aligned} \tag{5.67}$$

We remark that for improper torsion angles we use r_{jl} instead of r_{kl}. The analytical formulas are found by choosing a basis for $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$ and $\boldsymbol{d}$. Since $\boldsymbol{P}$ has a simple matrix representation in $\{\boldsymbol{u}_1,\boldsymbol{u}_2,\boldsymbol{u}_3\}$ and $\boldsymbol{Q}$ in $\{\boldsymbol{v}_1,\boldsymbol{v}_2,\boldsymbol{v}_3\}$, a good choice is:

$$\begin{aligned} \boldsymbol{a} &= a_1\boldsymbol{u}_1 + a_2\boldsymbol{u}_2 + a_3\boldsymbol{u}_3 \\ \boldsymbol{b} &= b_1\boldsymbol{u}_1 + b_2\boldsymbol{u}_2 + b_3\boldsymbol{u}_3 \\ \boldsymbol{c} &= c_1\boldsymbol{v}_1 + c_2\boldsymbol{v}_2 + c_3\boldsymbol{v}_3 \\ \boldsymbol{d} &= d_1\boldsymbol{v}_1 + d_2\boldsymbol{v}_2 + d_3\boldsymbol{v}_3. \end{aligned} \tag{5.68}$$

When we use these expressions in equation (5.66) and evaluate the operations, the final result is obtained

$$
\begin{aligned}
h_{1,r_{ij}} &= a_1,\ h_{1,\theta_{ijk}} = \frac{c_2}{r_{jk}} - \frac{a_2}{r_{ij}},\ h_{1,\tau_{ijkl}} = \frac{1}{r_{ij}\sin\theta_{ijk}}a_3 - \frac{\cos\theta_{ijk}}{r_{jk}\sin\theta_{ijk}}b_3 - \frac{\cos\theta_{jkl}}{r_{jk}\sin\theta_{jkl}}c_3 + \frac{1}{r_{kl}\sin\theta_{jkl}}d_3 \\
h_{2,r_{ij}} &= \frac{1}{r_{ij}}\left(a_2^2 + a_3^2\right),\ h_{2,\theta_{ijk}} = -2\left(\frac{c_1c_2}{r_{jk}^2} - \frac{a_1a_2}{r_{ij}^2}\right) + \frac{\cos\theta_{ijk}}{\sin\theta_{ijk}}\left(\frac{a_3^2}{r_{ij}^2} + \frac{b_3^2}{r_{jk}^2}\right) - 2\frac{a_3b_3}{r_{ij}r_{jk}\sin\theta_{ijk}} \\
h_{2,\tau_{ijkl}} &= \frac{2}{r_{ij}^2\sin^2\theta_{ijk}}\left(\cos\theta_{ijk}a_2a_3 - \sin\theta_{ijk}a_1a_3\right) \\
&\quad + \frac{2}{r_{ij}r_{jk}\sin^2\theta_{ijk}}\left\{-a_2b_3 - \cos^2\theta a_3b_2 + \cos\theta_{ijk}\sin\theta_{ijk}a_3b_1\right\} \\
&\quad + \frac{2}{r_{jk}^2\sin^2\theta_{ijk}}\left\{\frac{1}{2}\cos\theta_{ijk}\left(2+\sin^2\theta_{ijk}\right)b_2b_3 - \frac{1}{2}\sin^3\theta_{ijk}b_1b_3\right\} \\
&\quad + \frac{2}{r_{jk}^2\sin^2\theta_{jkl}}\left\{-\frac{1}{2}\left(1+\cos^2\theta_{jkl}\right)c_2c_3 + \cos\theta_{jkl}\sin\theta_{jkl}c_1c_3\right\} \\
&\quad + \frac{2}{r_{jk}r_{kl}\sin^2\theta_{jkl}}\left\{\cos\theta_{jkl}\left(c_2d_3 + c_3d_2\right) - \sin\theta_{jkl}c_3d_1\right\} - \frac{2}{r_{kl}^2\sin^2\theta_{jkl}}d_2d_3.
\end{aligned}
\tag{5.69}
$$

We see that only r_{ij}, r_{jk}, r_{kl} (or r_{jl}), θ_{ijk} and θ_{jkl} as well the components of $\boldsymbol{a}$ and $\boldsymbol{b}$ in the basis $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3\}$ and $\boldsymbol{c}$ and $\boldsymbol{d}$ in the basis $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3\}$ enter into these formulas.

5.5 Conclusion

We have formulated simple rules to calculate the second-order derivatives of a function of an internal coordinate. The method is based on expanding the block matrices that form the Hessian in terms of operators represented in two orthonormal bases.

We derived algorithms to generate the Hessian of a function of an internal coordinate. For a torsion-angle-dependent function, a calculation of three bond lengths, two bond angles and two orthonormal bases is required and it is numerically stable if the function is well-defined. The algorithms can be implemented with high efficiency in high-level programming languages based on vectorization. An efficient calculation of the Hessian opens the possibility of utilizing large-scale Hessian based optimization routines in geometry optimization that use directions of negative curvature and ensure that the optimized structure has a positive semidefinite Hessian[58, 59]. Furthermore, the curvature of the energy landscape and transition states may be studied more efficiently during a geometry optimization or a molecular dynamics simulation.

Finally, we used the formulas to derive a second-order expansion of an internal coordinate. These have very compact expressions, which for example can be used in line search routines or to calculate the expansion of an internal coordinate to second order. In future work, the usability of this method in geometry optimization and dynamics will be investigated.

5.6 Acknowledgements

The author thanks Peter Røgen for valuable discussions and suggestions.

Chapter 6

On the importance of the distance measures used to train and test knowledge-based potentials for proteins

M. Carlsen, P. Koehl, P. Røgen, On the importance of the distance measures used to train and test knowledge-based potentials for proteins , *PLOS ONE*, vol. 9, pp. e109335, 2014.

Abstract. Knowledge-based potentials are energy functions derived from the analysis of databases of protein structures and sequences. They can be divided into two classes. Potentials from the first class are based on a direct conversion of the distributions of some geometric properties observed in native protein structures into energy values, while potentials from the second class are trained to mimic quantitatively the geometric differences between incorrectly folded models and native structures. In this paper, we focus on the relationship between energy and geometry when training the second class of knowledge-based potentials. We assume that the difference in energy between a decoy structure and the corresponding native structure is linearly related to the distance between the two structures. We trained two distance-based knowledge-based potentials accordingly, one based on all inter-residue distances (PPD), while the other had the set of all distances filtered to reflect consistency in an ensemble of decoys (PPE). We tested four types of metric to characterize the distance between the decoy and the native structure, two based on extrinsic geometry (RMSD and GTD-TS*), and two based on intrinsic geometry (Q* and MT). The corresponding eight potentials were tested on a large collection of decoy sets. We found that it is usually better to train a potential using an intrinsic distance measure. We also found that PPE outperforms PPD, emphasizing the benefits of capturing consistent information in an ensemble. The relevance of these results for the design of knowledge-based potentials is discussed.

6.1 Introduction

Proteins are the essential macromolecules inside cells that perform nearly all cellular functions. Just like macroscopic tools, their shapes is a key feature for defining their functions. Structural biologists have embarked upon the challenge of finding the structures of all proteins, in hopes of unraveling this relationship between geometry and biological activity and learn in the process how cells function. Determining experimentally the structure of a protein at the atomic level however is not yet an easy task: this can be indirectly deduced from the fact that we currently know millions of protein sequences but less than hundred thousand protein structures. Predicting the structure of a protein from first principles is not much easier: direct applications of the ideas that have been used for modeling small molecules have not yet been successful on these much larger molecules. Recent reports on the advancements of *ab initio* techniques clearly show that the protein structure prediction community is making progress, but that the quality of the models they generate do not meet yet the stringent accuracy requirements to become useful to the biologists [4]. Interestingly, the series of Critical Assessment of protein Structure Prediction (CASP) meetings have highlighted that while the methods for generating models of protein structures have improved significantly[81], identifying the native-like conformations among the large collections of model structures (also called decoys) remains a significant challenge[82, 83]. In this paper we focus on this problem.

Anfinsen's thermodynamics hypothesis states that the native structure of a protein is determined only by its amino acid sequence [1]. Structural and computational biologists translate this postulate into the statement, that under physiological conditions, the native state of a protein is a unique, stable minimum of the free energy. The key to solving the protein structure prediction problem amounts therefore to finding an accurate representation of this free energy function and several methods have been proposed to construct reasonable approximations of it. The two most common approaches rely on semiempirical and statistical potentials, respectively. Semiempirical methods are derived from knowledge of the basic physical principles whereas statistical potentials are based on the nonrandom statistics of known protein structures [84]. Statistical energy functions are either residue based or atom based and the most recent statistical potentials include pairwise interactions, orientations of side-chains[85], secondary structural preferences, solvent-exposure, and other geometric properties of proteins [86]. We note that there have been attempts to combine physics-based and statistics-based potentials to improve protein structure refinement [26, 87, 27, 88, 28].

Current protein structure prediction methods require potentials that ideally should assign "scores" to a protein structure model such that the higher the score, the less native-like the model is, where native-like is measured in terms of a distance d from the model to the native structure. If this condition is satisfied then the potential is expected to detect near native conformations even when the native conformation is not present; in addition, such an ideal potential could then be used for model refinement. In mathematical terms this can be expressed as the score function f satisfying

$$f\left(seq_i, \boldsymbol{r}_i + \boldsymbol{dr}\right) = f\left(seq_i, \boldsymbol{r}_i\right) + d\left(\boldsymbol{r}_i, \boldsymbol{r}_i + \boldsymbol{dr}\right), \tag{6.1}$$

for any sequence seq_i and all deformations $\boldsymbol{dr}$ of its native structure $\boldsymbol{r}_i$.

Several methods have been developed to optimize potentials towards this goal[89, 90, 91, 92]. The choice of the distance measure d is critical to the success of these

methods. The standard distance measure when comparing protein structural models is RMSD, i.e. the root mean square distance between the two models after optimal translation and rotation. RMSD however has been replaced in recent CASP experiments by the global distance test (GDT-TS[10]) due to its undesirable sensitivity towards local changes in a protein structure; GDT-TS has become one of the most commonly used distance measures in protein structure prediction. A less commonly used distance measure is the fraction of known native contacts, Q. Q quantifies the changes in the number of "contacts" found in the native structure compared to the model structure that is evaluated, where a contact corresponds to two residues being within a given threshold distance from each other. All the distance measures mentioned above identify geometric differences between two structural models but do not attempt to assess if these differences could be assigned to fluctuations due to the dynamics of the protein. Such differences would be less of a concern if they were related to geometric differences that can be explained by dynamics. As an attempt to identify the role of dynamics, Perez *et al.* recently introduced FlexE, a method based on a simple elastic network model that uses the deformation energy as a measure of the similarity between two structures [11]. As such, FlexE is expected to distinguish biologically relevant conformational changes from random changes.

In this work, we investigate the importance of the distance function d when optimizing an energy function f towards satisfying equation 6.1. We train two new $C\alpha$-based pairwise potentials, PPD and PPE, to mimic the distance between the model structure considered and its corresponding native structure, using four different definitions of the distance measure, namely RMSD, GDT-TS, Q, and MT, where MT is an anharmonic version of FlexE. These energy functions are trained and tested on sets extracted from the high resolution decoy dataset Titan-HRD[47], as well as on well known decoy datasets from DecoysRUs [93] and Rosetta [94]. We have also analyzed the performance of our potentials on the server generated Stage_1 and Stage_2 decoy sets from CASP 10 [95].

The paper is organized as follows. The next section introduces the different distance measures and describes our procedures for training and testing the potentials PPD and PPE. The following section shows the results on different decoy sets as well as a comparison between PPD, PPE, two statistical knowledge-based potentials and a semi-empirical physical potential. We conclude with a discussion of the importance of the choice of the distance measure and describe potential future work.

6.2 Materials and Methods

6.2.1 Geometrical distances between two structural models of the same protein

Let us consider two structural models A and B of the same protein P with N amino acids. We represent the two models as discrete sets of N points, $A = (a_1, a_2, \ldots, a_N)$ and $B = (b_1, b_2, \ldots, b_N)$ where the points a_i and b_i correspond to the positions of the $C\alpha$ atoms i in the two structures. We assume that the correspondence table between A and B is known and set such that a_i corresponds to b_i for all $i \in [1, N]$. We measure the distance between the two models either based on the Euclidean distance between the two sets of points (RMSD and GDT-TS), on differences between contact maps within each set (Q), or on an elastic network (MT).

RMSD, i.e. root mean square deviation, is the Euclidean distance between the corresponding points a_i and b_i after one of the two sets of points (usually set B) has been optimally transformed by a rigid body transformation G:

$$RMSD = \min_G \sqrt{\frac{\sum_{i=1}^{N} \|a_i - G(b_i)\|^2}{N}}. \tag{6.2}$$

The rigid body transformation G is a transformation that does not produce changes in the size, shape, or topology of the protein. Such transformations are compositions of rotations and translations. Many closed-form solutions to the problem of finding the optimal G have been derived [7, 8, 9]. We note that RMSD as defined above is a metric [96].

RMSD is a distance measure based on the L_2 norm; as such, it is highly sensitive to outliers, for example due to the presence of large albeit local differences between the two structures. The global distance test (GDT) was developed to decrease this sensitivity [10]. GDT focuses on the regions of the structures that can be correctly aligned by counting the number of residues that can be superimposed within a given cutoff distance. GDT-TS (where TS stands for Total Score), combines this information for multiple cutoffs:

$$GDT - TS = \frac{n_1 + n_2 + n_4 + n_8}{4n}, \tag{6.3}$$

where n_1, n_2, n_4, and n_8 are the numbers of aligned residues within 1, 2, 4, and 8 Ångströms, respectively, and n is the total aligned length. Note that GDT-TS is a quantity between 0 and 1 that represents similarity, with low values corresponding to bad correspondences, and high values (close to or equal to 1) indicating that the two models are highly similar. We have converted this similarity measure into a distance by considering GDT-TS* = 1-GDT-TS.

RMSD and GDT-TS* are computed after the two model structures have been optimally superposed. An alternative approach is to consider the intrinsic geometry of the two structures, as captured for example by a distance matrix that contains all $C\alpha - C\alpha$ distances internal to one structure. Q and MT are two examples of distance measures that use this alternate approach.

The fraction of native contacts, Q, is a distance measure that quantifies the changes of a contact map between two models for the same structure. A contact map is usually defined as

$$S_{i,j} = \begin{cases} 1 & \text{if residues } i \text{ and } j \text{ are in contact} \\ 0 & \text{otherwise ,} \end{cases}$$

where two residues are in contact if they are within a given distance threshold. In this paper, we set this threshold to 9 Å. Q is then defined by

$$Q = \frac{sc}{sc + lc},$$

where sc is the number of shared contacts and lc is the number of lost contacts. Just like GDT-TS, Q is a measure of similarity. We convert it into a distance measure by defining Q*=1-Q.

Q* quantifies changes in the contact map of a structure with no consideration of what could have been the reasons for these changes. FlexE is a new measure of similarity between protein structures that was introduced as an attempt to distinguish those changes that are biologically relevant [11]. It is based on the concept of elastic network that assigns virtual isotropic springs between pairs of residues. Elastic network models are used in normal mode analysis[97, 98] for example to reconstruct proteins[99], to generate decoy sets[100], or to investigate thermal fluctuations about the native or equilibrium structure[101, 102]. In the formalism introduced by Perez et al[11], the distance measure FlexE between two structures N and D is assimilated to the energetic cost of deforming one of the structures into the other:

$$FlexE(N,D) = \frac{1}{N_{\text{res}}} \sum_{i,j=1}^{N_{\text{res}}} S_{i,j}^{N} k_{ij} \left(r_{ij}^{N} - r_{ij}^{D}\right)^2 , \tag{6.4}$$

where N_{res} is the number of residues in N and D, $S_{i,j}^{N}$ is a contact map for structure N, r_{ij}^{N} and r_{ij}^{D} are the distances between the $C\alpha$ atoms of residues i and j in structures N and D, respectively, and k_{ij} is a force constant associated to the link between i and j. In our implementation of FlexE, we set all force constants to 1. We modify the quadratic term in equation 6.4 with a term congruent to the potential introduced by Toda [103] to study chains of particles interacting with non-linear forces.

The corresponding variant of FlexE, which we name MT, is defined as:

$$MT(N,D) = \frac{1}{N_{\text{res}}} \sum_{i,j=1}^{N_{\text{res}}} \frac{S_{i,j}^{N}}{b^2} \left(e^{-(r_{ij}^{D} - r_{ij}^{N})b} + \left(r_{ij}^{D} - r_{ij}^{N}\right) b - 1\right), \tag{6.5}$$

where b is a parameter which we set to 0.5. We note that MT is equal to FlexE for small perturbations of the distances between residues; for large perturbations however, it penalizes compression more than extension. Finally the use of the fixed native contact map for all native-decoy comparisons ensures that both FlexE(N,D) and MT(N,D) are well-defined.

6.2.2 Two new parametric potentials

A smooth, pairwise potential, PPD. We design a smooth knowledge based residue pair potential as done in [12]. For each of the 210 pairs of amino acids types we assume a potential that is determined by the corresponding Cα-Cα distance. We model the interaction as a uniform cubic b-spline with compact support within 1 Å to 12 Å and 8 degrees of freedom, see e.g. [104]. With this model an interaction tends smoothly to zero energy at distances greater than 12 Å and is modeled freely within 4 Å -9 Å. The pair potential has $8 \times 210 = 1680$ parameters in total. The corresponding potential, PPD, is defined as

$$PPD = \sum_{i<j} \sum_{p} C_{p}^{aa(i)aa(j)} B_p(r_{i,j}) , \tag{6.6}$$

where aa$(i) \in \{1, \ldots, 20\}$ is the amino acid type of the i-th residue and $B_p(r_{i,j})$ is the p-th b-spline basis function evaluated on the distance between the i-th and j-th residues. $C_{p}^{aa(i)aa(j)}$ are the model parameters determined by the optimization procedure described below.

A consensus potential, PPE. We introduce a novel smooth ensemble based pair potential (PPE) that forms an artificial funnel relative to a pre-calculated contact map:

$$PPE = \sum_{i<j} S_{i,j} \sum_{p} C_p^{aa(i)aa(j)} B_p(r_{i,j}) \ , \tag{6.7}$$

where $S_{i,j}$ is an consensus contact map. The method to calculate the consensus contact map is described below. It is based on a similar consensus method that constructs the reference contact map from an ensemble of decoys[105].

A consensus contact map. We introduce an iterative method to compute a consensus contact map of an ensemble of decoys. The first step is to construct a contact map from the most common contacts in the ensemble. Let $M_{i,j}$ be the fraction of contacts in the ensemble for the i,j-th residue pair. The contact map is then calculated as

$$S_{i,j} = \begin{cases} 1 & \text{if } M_{i,j} > \mu \\ 0 & \text{otherwise} \end{cases} \tag{6.8}$$

where μ is a cut-off fixed at 0.25. At each step, we select the 25% closest decoys to this contact map, where "closest" refers to the Hamming-distance to the contact map. This leads to a reduced ensemble from which a new contact map is computed, and the procedure is iterated. The algorithm usually converges in a few steps.

6.2.3 Optimizing the potentials

We design an energy landscape using a sculpting procedure. We assume that we possess a set of natives structures $\{N_i\}$ and that a set $\{D_{i,j}\}$ of decoy structures is known for each of these native structures. Let $\Delta E_{i,j}$ be the energy difference between the i-th native structure, N_i, and its j-th decoy, $D_{i,j}$, and let $d(N_i, D_{i,j})$ be the corresponding distance between N_i and $D_{i,j}$. Our method for optimizing a statistical potential [12] attempts to establish a funnel-shaped energy function by calculating the parameters that minimizes the sum of squared errors between $\Delta E_{i,j}$ and $\alpha_{N_i} d(N_i, D_{i,j})$ where α_{N_i} is a constant of proportionality. The problem can be stated as a quadratic programming (QP) problem with affine constraints,

$$\begin{aligned} \underset{\boldsymbol{X}, \alpha_1 \ldots \alpha_M}{\text{minimize}} \quad & \sum_{i,j} \|\Delta E_{i,j}(\boldsymbol{X}) - \alpha_{N_i} d(N_i, D_{i,j})\|^2 + \beta \|\boldsymbol{X}\|^2 \\ \text{subject to} \quad & 0.25 \leq \alpha_{N_i} \leq 4, \ \text{ for } \ i = 1 \ldots M \\ & \sum_i \alpha_{N_i} = M \ , \end{aligned} \tag{6.9}$$

where β is a fixed parameter used for regularization. The variables in this QP problem are $\boldsymbol{X}$, i.e. the vector of coefficients $C^{i,j}$ introduced above, and the constants of proportionality $\alpha_{N_1} \ldots \alpha_{N_M}$, where M is the number of proteins in the training set. The last term $\beta \|\boldsymbol{X}\|^2$ is a regularization term that adds a penalty onto the modulus of $\boldsymbol{X}$. The preprocessing is trivially parallelizable since each of the terms, $\|\Delta E_{i,j}(\boldsymbol{X}) - \alpha_i d(N_i, D_{i,j})\|^2$, can be calculated individually. As a consequence, the QP requires little memory and is fast to compute. We use the optimization package cplex to solve it.

6.2.4 Training and test sets

It is a nontrivial task to construct a "good" set of decoy structures. Any such decoy set relies on a sampling of the conformational space accessible to the protein structure of interest. The specific techniques used to generate such sampling are prone to biases [106], leading to poor sampling of the corresponding free energy surfaces. These approximate energy surfaces may not adopt a funnel like geometry in the neighborhood of the native structure and may contain many artificial potential energy barriers. To avoid the risk of learning from a specific bias introduced by one sampling technique, we have considered a variety of test sets to train and measure the performances of our energy functions. Of particular interest to us are near-native test sets since we design energy functions to mimic the neighborhoods of native structures.

We have chosen part of the Titan High Resolution Decoy set[47] as our training set. The list of proteins included in this set was originally proposed by Zhou and Skolnik [92]; it was selected on the basis that it is composed of a representative set of nonhomologous single domain proteins with maximum pairwise sequence similarity reported to be 35% . The models included in the decoy sets were generated using the torsion angle dynamics program DYANA[107] subject to distance constraints that are set to preserve the hydrophobic core of a protein. It is assumed that the hydrophobic core includes all residues within a β strand as well as all hydrophobic residues within an α-helix. The set includes 1400 proteins in total (compared to 1489 proteins in the original set of Zhou and Skolnik [92]). We eliminated all short proteins with a large radius of gyration as these proteins are overfitted by the optimization and are usually separate stretched secondary structures. We divided the remaining proteins into a training set of 1155 proteins with an average of 994 decoys per native structure (Titan-HRD*) and a test set of 142 proteins with an average of 854 decoys per native structure (Titan-HRD). The average GDT-TS distances between native and decoys over the training and test sets are 0.75 and 0.76 with a mean absolute deviation of 0.1, respectively. Note that we will use the mean absolute deviation (the l_1-norm) instead of the standard deviation (the l_2-norm) as it puts less weight on outliers.

Apart from the Titan-HRD set we use 10 freely available decoy sets that were generated using different procedures. These include 6 sets taken from DecoysRUs[93] (4 state reduced[108], hg structal[93], fisa[109], fisa casp3[109], lmds[110] and lattice ssfit[111, 112]). We also included two older versions of the Rosetta decoy sets (Rosetta-All[113] , Rosetta-Tsai[94]), the newest version Rosetta-Baker available at http://depts.washington.edu/bakerpg/decoys/ and the I-Tasser Set II[114].

The different CASP meetings have highlighted successes and failures in generating model structures that resemble the native structures of proteins. A repository of all models that have been proposed as answers to the prediction challenges that were part of these meetings is available on the CASP web page (`http://predictioncenter.org`). This repository provides a wealth of information on protein structure modeling, as well as useful test cases to assess the quality of new potential energy functions. We have therefore considered five CASP sets each containing models predicted by a variety of methods from the different CASP meetings (302 ensembles in total). We also generated CASP-HRD, a high resolution decoy subset of CASP 5 - 9, which includes models that have a TM score [115] larger than 0.5 and a RMSD less than 4 Å to the native structures. This cutoff was chosen based on the observation made by Xu and Zhang, which states that two decoys belong to the same fold when their TM-score to a native structure is higher than 0.5[22]. CASP-HRD is constructed to have nearly the same

average distance measure value as Titan-HRD but we find smaller variations of the distance measures for CASP-HRD. In that sense, it does include variations with different structural characteristics compared to Titan-HRD as it is generated by many different methods, while Titan-HRD is more homogeneous.

The total number of ensembles excluding Titan-HRD, Titan-HRD*, and CASP-HRD is 546 with an average GDT-TS between its decoys and their corresponding native structures of 0.47 with a average mean absolute deviation of 0.16. We refer to this set as "Test Set All" (TSA).

Finally, we include decoys from the latest CASP experiment, CASP10. A critical component of the CASP experiment is the assessment of the predictions that are submitted as putative models for the target proteins considered. This assessment is performed by the CASP assessors but also by the CASP community, with considerable enthusiasm, as observed in CASP10 [95]. The procedure for assessing the predictions in CASP10 differed from that of previous CASPs. The main difference was the introduction of two stages, labeled Stage_1 and Stage_2. For the former, twenty of the supposedly best predictions for each CASP target were released for assessment. Subsequently, hundred and fifty decoys were released for each target, defining Stage_2. Stage_1 ensembles are designed to survey single model assessment methods, while stage_2 allows for the survey of methods that rely on ensembles for the assessment of models. We have considered 93 targets from CASP10 for which both Stage_1 and Stage_2 test sets are available from the CASP web site (http://www.predictioncenter.org/casp10/). Compared to the other decoy sets described above, these sets contain longer protein chains. The models they include are usually as distant from their native counterparts as observed for the datasets from the previous CASP meetings. These sets however are more compact, i.e. with less diversity in distances, especially for the Stage_2 sets that resemble the CASP-HRD sets in that respect.

In table 6.1, we report the mean characteristics of these decoy sets (size, diversity, ...) as well as information about their availability.

Preprocessing the decoy sets. To guarantee that the decoys included in a set are consistent in length with their corresponding native structure, we performed the following two-step preprocessing. First, we removed all residues in the decoys with missing backbone atoms (Cα, N, C, and O). Second, we extracted the sequences from the decoy structure files and aligned these sequences with the native sequence of the protein of interest (where the native sequence is derived from the ATOM record in the corresponding PDB file). If these alignments include trailing unmatched residues either in the decoys or in the native structure, these residues are removed until all sequences are identical. We found that this procedure was necessary for some of the decoy sets described above.

6.2.5 Assessing the quality of decoy selection: R-score

Given a distance measure and an energy function, an ensemble of decoy protein conformations contains a "best" distance model, i.e. the conformation that is closest geometrically to the native structure, as well as a "best" energy model, i.e. the model whose energy is the lowest. Ideally, these two "best" models should be the same; in practice however, they are different due to shortcomings of the potential energy function. To quantify this difference we introduce the R-score as follows. Let $\mathcal{D}$ be the ensemble

Table 6.1: Properties of the different protein decoy sets used in this study

Decoy set	Nprot [h]	Nres [h]	Ndecoys [h]	RMSD	MT	GDT-TS	Q
Titan-HRD [a]	142	127 (35)	854 (119)	2.4 (0.5)	2.7 (1)	0.76 (0.1)	0.85 (0.04)
Titan-HRD* [a]	1155	111 (35)	994 (138)	2.6 (0.6)	2.7 (1)	0.75 (0.1)	0.85 (0.04)
TASSER Set II [b]	55	80 (17)	438 (98)	6.3 (1.5)	9.3 (3.2)	0.54 (0.05)	0.77 (0.03)
hg Structal [c]	28	150 (7)	29 (0)	4.1 (1.2)	4.4 (1.5)	0.71 (0.07)	0.85 (0.04)
4-state [c]	7	64 (4.9)	664 (15)	5.2 (1.4))	8.3 (2.9)	0.53 (0.11)	0.75 (0.05)
fisa [c]	4	60 (10)	500 (0.4)	7.5 (1.8)	8.6 (1.7)	0.47 (0.06)	0.75 (0.06)
fisa CASP3 [c]	5	88 (15)	1437 (390)	12 (1.6)	21 (4.1)	0.3 (0.03)	0.67 (0.02)
lmds [c]	10	53 (10)	433 (79)	7.7 (1.1)	12 (2.6)	0.46 (0.04)	0.72 (0.03)
lattice ssfit [c]	8	71 (10)	1997 (1.5)	9.9 (1.0)	17 (2.4)	0.3 (0.03)	0.64 (0.02)
Rosetta-All [d]	41	82 (25))	999 (0.5)	12 (1.4)	29 (5.6)	0.27 (0.03)	0.61 (0.02)
Rosetta-Tsai [d]	29	63 (9.4)	1862 (43)	7.4 (2.1)	11 (3.9)	0.46 (0.08)	0.73 (0.04)
Rosetta-Baker [d]	57	88 (20)	100 (0)	8.5 (1.4)	15 (3.3)	0.45 (0.05)	0.76 (0.03)
CASP5 [e]	41	202 (78)	117 (41)	13 (3.7)	29 (14)	0.38 (0.12)	0.68 (0.08)
CASP6 [e]	39	172 (71)	216 (34)	13 (4.9)	27 (16)	0.39 (0.12)	0.70 (0.08)
CASP7 [e]	64	183 (80)	349 (40)	10 (3.4)	17 (10)	0.47 (0.11)	0.75 (0.07)
CASP8 [e]	77	187 (81)	334 (67)	8.8 (3.1)	13 (8.6)	0.54 (0.11)	0.79 (0.06)
CASP9 [e]	81	180 (81)	402 (95)	11 (4.9)	19 (14)	0.49 (0.12)	0.77 (0.07)
CASP-HRD [e]	109	188 (79)	192 (72)	2.8 (0.4)	2.2 (0.6)	0.76 (0.03)	0.89 (0.02)
CASP10-stage1 [f]	93	232 (102)	18 (1.9)	13 (4.3)	20 (9.4)	0.46 (0.08)	0.76 (0.05)
CASP10-stage2 [f]	93	232 (102)	132 (7.6)	11 (3.7)	17 (8.2)	0.55 (0.03)	0.80 (0.03)
TSA TM > 0.5 [g]	242	179 (77)	291 (119)	6.3 (2.67)	9.4 (5.5)	0.63 (0.09)	0.82 (0.05)
TSA TM < 0.5 [g]	303	110 (48)	602 (436)	12 (3.9)	23 (12)	0.34 (0.1)	0.68 (0.07)

[a] Training set (Titan HRD) and test set (Titan HRD*) from the Titan High resolution decoy set[47], available at `http://titan.princeton.edu/2010-10-11/Decoys/`.

[b] Tasser Set II is a structurally non-redundant set of protein structures and decoys derived with the program TASSER. It is available at `http://zhanglab.ccmb.med.umich.edu/decoys/`.

[c] Decoy sets from the Decoys 'R' us repository `http://dd.compbio.washington.edu`.

[d] Different decoy Rosetta-based decoy sets (see text for details), available at `http://depts.washington.edu/bakerpg/decoys/`.

[e] Collection of models from the successive CASP5 to CASP9 experiments, available from the CASP web site `http://predictioncenter.org`. CASP-HRD is a high resolution subset of the union of the five sets CASP5 to CASP9, which includes models that have a TM-score larger than 0.5 and a RMSD less than 4 Å to the native structures.

[f] The Stage_1 and Stage_2 decoy sets used in the CASP10 quality assessment category, available from the CASP web site `http://predictioncenter.org`. For details on how these sets are prepared, see [95].

[g] All high and low resolution targets (TSA TM-score > 0.5)/(TSA TM-score < 0.5) are listed in the files S_TSAh and S_TSAl respectively found in the supporting information.

[h] Nprot is the number of different proteins in the dataset, Nres is the average number of residues computed over all proteins in a dataset, and Ndecoys is the average number of decoys per proteins, averaged over the dataset. RMSD, MT, GDT-TS, and Q are the distance measures between the decoys and the corresponding native structures, averaged over all decoys and all proteins. We provide both the average values and the average mean absolute deviations (in parenthesis).

of decoys and let X_i be one of its elements. The corresponding native structure is N. We define the mapping S_d from $\mathcal{D}$ to $\mathbb{R}$ as $S_d(X_i) = d(X_i, N)$, i.e. the distance between the decoy X and N, where d can be any of the four distance measures defined above. We name X_E the decoy with the lowest energy, i.e. $E(X_E) \leq E(X) \quad \forall X \in \mathcal{D}$. In parallel, we name X_d the decoy closest to N with respect of the distance d, i.e. $S_d(X_d) \leq S_d(X) \quad \forall X \in \mathcal{D}$. The R score for d and E is defined as:

$$R(d, E) \equiv \begin{cases} \frac{S_d(X_E) - \langle S_d \rangle}{S_d(X_d) - \langle S_d \rangle} & \text{if } |S_d(X_E) - \langle S_d \rangle| \leq |S_d(X_d) - \langle S_d \rangle| \\ -1 & \text{otherwise} \end{cases}, \tag{6.10}$$

where $\langle S_d \rangle$ is the average value for S_d over the decoy set $\mathcal{D}$. $R(d, E)$ is designed to assess how well E mimics S in finding the best decoy. It takes values between -1 and 1 where 1 indicates that the energy has picked the best decoy. We fix the lower limit at -1 to avoid having outliers being assigned very low negative values. Note, that if an ensemble does not contain outliers then 0 is the random expectation. If we furthermore assume that the distances $S_d(X)$ are uniformly distributed then $(1 - R(d, E))/2$ is the fraction of decoys with a distance to the native structure better than $S_d(X_E)$. The R score can also be seen as the ratio between the Z-score of the best energy model, $(S_d(X_E) - \langle S_d \rangle)/\sigma(S_d)$, and the Z-score of the best distance model, $(S_d X_d - \langle S_d \rangle)/\sigma(S_d)$, where $\sigma(S_d)$ is the standard deviation for S_d over the decoy set $\mathcal{D}$.

6.2.6 Assessing how well the energy functions mimic a funnel in the neighborhood of the native structure

To measure how far the energy E is from the desired linear funnel shape given by Equation 6.1 relative to the distance measure d we report the Pearson's correlation coefficient $Corr(d, E)$ between the energy values $E(X_i)$ and distance measures $S_d(X_i)$ over all decoys X_i in the decoy set:

$$Corr(d, E) = \frac{1}{N-1} \sum_{i=1}^{N} \frac{S_d(X_i) - \langle S_d \rangle}{\sigma(S_d)} \frac{E(X_i) - \langle E \rangle}{\sigma(E)}, \tag{6.11}$$

where $\langle . \rangle$ and $\sigma(.)$ stand for the mean and standard deviation over the decoy set considered.

6.2.7 Comparing two distance measures d_1 and d_2

In the two previous subsections, we have defined a R-score $R(d, E)$ and a correlation coefficient $Corr(d, E)$ to measure how well an energy function E mimics a distance measure d. Both quantities can be used as is to compare two distance measures d_1 and d_2. Indeed, d_2 can be assimilated to a pseudo energy function, akin to the definition of FlexE given in equation 6.4. The R-score and correlation coefficient between d_1 and d_2 are then simply $R(d_1, d_2)$ and $Corr(d_1, d_2)$, respectively. $Corr(d1, d2)$ measures the dependence between $d1$ and $d2$ over a decoy set, while $R(d1, d2)$ checks the "quality" of the best decoy identified by d_2, as measured by d_1. Note that this R-score between distance measures may not be symmetric.

6.3 Results and Discussion

6.3.1 The diversity of the distance measures

There is no unique way to compare three dimensional shapes. When comparing protein structures, two main classes of distance measures have been proposed, those based on a Euclidean distance between the positions of the atoms of the two proteins (after proper translation and rotation of one of them), and those based on the intrinsic geometry of the structures. We have considered two examples in each class, namely RMSD and GDT-TS* for the former, and MT and Q* for the latter. A full description of these four distance metrics is given in Material and Methods. As these measures capture changes of different geometric properties of the protein structures, there is no reason to believe that they are equivalent. To test the degrees to which these distances differ, we have compared them on three different sets of decoys, namely Titan-HRD, CASP-HRD, and TSA, using two different report scores, $Corr$ and R, where $Corr$ is the Pearson's correlation coefficient that measures how well d_1 mimics d_2 over a large range of distance values while R measures how (metrically) wrong the best candidate of one distance measure (i.e. the decoy with the smallest distance to its corresponding native structure) is when measured by another distance (see Materials and Methods for details). Results for $Corr$ and R are given in tables 6.2 and 6.3, respectively.

The correlations between the distance measures are high on the Titan-HRD set of decoys, with values above 0.87 for the correlation coefficients. The corresponding R-scores are above 0.76. If we assume uniform distributions of the native-decoy distances over a decoy set, the best decoy by one distance measure on average is ranked within the top 5% and within the top 12% by another distance measure for R scores of 0.9 and 0.76, respectively. These high scores are expected, as the Titan-HRD decoys are high resolution, usually very close to their native structure counterparts (see Table 1). It is interesting however that the R score between RMSD and Q* is relatively low (0.76), even on this high resolution data set. This low value indicates that a "good" decoy defined by Q* may explore a range of RMSD values. In contrast, a decoy that is close to the native structure with respect to RMSD usually has a high percentage of native contacts, as highlighted by the R score between Q* and RMSD of 0.87. In fact, we observe that the best RMSD decoy is generally scored better by the three other distance measures.

While CASP-HRD also contains high resolution decoys that are close to their corresponding native structures (with RMSD $<$ 4 Å and TM scores above 0.5), the four distance measures we tested are less dependent on this dataset than on Titan-HRD, both globally as scored by correlation coefficients and locally (i.e. in picking a "best" decoy), as highlighted by the R scores. We see two possible reasons for these differences between the two groups of decoy sets. First, the decoys in Titan-HRD are homogeneous, as they all contain the same hydrophobic cores as the native structures. In contrast, the CASP decoys were derived with many different methods, leading to heterogeneity in their geometry. Second, we cannot exclude an effect of sample size, as on average the sets included in Titan-HRD contain four times more decoys and larger average mean absolute deviation of distance measures than the sets included in CASP-HRD (see Table 1).

TSA, which stands for "Test Sets All" is a large heterogeneous collection of decoy sets that were generated by many different techniques (see Materials and methods for

Table 6.2: Correlations between the four distance measures

		Distance d_2			
Test set	Distance d_1	RMSD	MT	GDT-TS*	Q*
Titan-HRD	RMSD	1[a]	0.92 (0.06)	0.92 (0.04)	0.87 (0.08)
	MT	0.92 (0.06)	1	0.92 (0.03)	0.94 (0.03)
	GDT-TS*	0.92 (0.04)	0.92 (0.03)	1	0.95 (0.03)
	Q*	0.87 (0.08)	0.94 (0.03)	0.95 (0.03)	1
CASP-HRD	RMSD	1	0.74 (0.16)	0.73 (0.14)	0.6 (0.19)
	MT	0.74 (0.16)	1	0.72 (0.13)	0.83 (0.07)
	GDT-TS*	0.73 (0.14)	0.72 (0.13)	1	0.74 (0.13)
	Q*	0.6 (0.19)	0.83 (0.07)	0.74 (0.13)	1
CASP10-stage1	RMSD	1	0.83 (0.16)	0.71 (0.24)	0.68 (0.24)
	MT	0.83 (0.16)	1	0.73 (0.2)	0.82 (0.14)
	GDT-TS*	0.71 (0.24)	0.73 (0.2)	1	0.86 (0.12)
	Q*	0.68 (0.24)	0.82 (0.14)	0.86 (0.12)	1
CASP10-stage2	RMSD	1	0.78 (0.16)	0.51 (0.22)	0.49 (0.19)
	MT	0.78 (0.16)	1	0.52 (0.2)	0.69 (0.14)
	GDT-TS*	0.51 (0.22)	0.52 (0.2)	1	0.64 (0.17)
	Q*	0.49 (0.19)	0.69 (0.14)	0.64 (0.17)	1
TSA	RMSD	1	0.92 (0.06)	0.8 (0.15)	0.82 (0.11)
	MT	0.92 (0.06)	1	0.78 (0.14)	0.85(0.08)
TM-score > 0.5	GDT-TS*	0.8 (0.15)	0.78 (0.14)	1	0.89 (0.12)
	Q*	0.82 (0.11)	0.85 (0.08)	0.89 (0.12)	1
TSA	RMSD	1	0.8 (0.12)	0.59 (0.24)	0.56 (0.18)
	MT	0.8 (0.12)	1	0.54 (0.2)	0.68(0.14)
TM-score < 0.5	GDT-TS*	0.59 (0.24)	0.54 (0.2)	1	0.67 (0.22)
	Q*	0.56 (0.18)	0.68 (0.14)	0.67 (0.22)	1

[a] Pearson's correlation coefficient $Corr(d_1, d_2)$ between the two distance measures d_1 and d_2. We provide both the average value and the mean absolute deviation (in parenthesis) over the data set considered.

details). Some of these decoy sets are high-resolution, i.e. contains mostly native-like structures, while others are more diverse, containing decoys that are very different from their corresponding native structures, both in terms of secondary structure content and three-dimensional organization. To assess the importance of this diversity, we selected within the TSA group of decoy sets two subgroups, those for which the decoys have average TM score larger than 0.5, and those with average TM score smaller than 0.5. This 0.5 cutoff was again chosen based on the observation made by Xu and Zhang that two decoys belong to the same fold when their TM-scores to a native structure is higher than 0.5[22]. Table 1 shows that TSA TM-score > 0.5 generally contain longer chains with fewer decoys when compared to the TSA TM-score < 0.5 set. The two sets are fully listed in the files S_TSAh and S_TSAl found in the supporting information. Tables 6.2 and 6.3 show that the distance measures behave on the high-resolution subgroup (TM > 0.5) as on the Titan-HRD test set, i.e. with high correlations and high R scores, meaning that they are very similar to each other. On the low-resolution subgroup (TM

Table 6.3: Comparing the best models picked by different distance measures

		Distance d_2			
Test set	Distance d_1	RMSD	MT	GDT-TS*	Q*
Titan-HRD	RMSD	1[a]	0.88 (0.12)	0.91 (0.09)	0.76 (0.17)
	MT	0.94 (0.06)	1	0.92 (0.08)	0.91 (0.07)
	GDT-TS*	0.96 (0.04)	0.94 (0.07)	1	0.91 (0.08)
	Q*	0.87 (0.09)	0.92 (0.07)	0.89 (0.09)	1
CASP-HRD	RMSD	1	0.71 (0.26)	0.79 (0.22)	0.49 (0.38)
	MT	0.76 (0.22)	1	0.76 (0.22)	0.76 (0.23)
	GDT-TS*	0.8 (0.22)	0.68 (0.27)	1	0.48 (0.39)
	Q*	0.57 (0.33)	0.81 (0.16)	0.66 (0.24)	1
CASP10-stage1	RMSD	1	0.81 (0.24)	0.75 (0.31)	0.79 (0.23)
	MT	0.9 (0.13)	1	0.85 (0.19)	0.94 (0.09)
	GDT-TS*	0.79 (0.24)	0.78 (0.24)	1	0.82 (0.2)
	Q*	0.78 (0.22)	0.88 (0.14)	0.8 (0.23)	1
CASP10-stage2	RMSD	1	0.76 (0.22)	0.71 (0.3)	0.63 (0.29)
	MT	0.83 (0.18)	1	0.73 (0.24)	0.83 (0.19)
	GDT-TS*	0.73 (0.26)	0.65 (0.24)	1	0.59 (0.29)
	Q*	0.62 (0.29)	0.82 (0.18)	0.62 (0.23)	1
TSA	RMSD	1	0.9 (0.11)	0.84 (0.19)	0.81 (0.18)
	MT	0.94 (0.07)	1	0.88 (0.14)	0.92 (0.09)
TM-score > 0.5	GDT-TS*	0.85 (0.16)	0.79 (0.21)	1	0.73 (0.24)
	Q*	0.79 (0.18)	0.89 (0.11)	0.81 (0.16)	1
TSA	RMSD	1	0.83 (0.19)	0.73 (0.27)	0.71 (0.27)
	MT	0.87 (0.14)	1	0.74 (0.27)	0.88 (0.14)
TM-score < 0.5	GDT-TS*	0.74 (0.27)	0.7 (0.27)	1	0.67 (0.27)
	Q*	0.68 (0.27)	0.85 (0.16)	0.68 (0.27)	1

[a] R-score $R(d_1, d_2)$ between the two distance measures d_1 and d_2. We provide both the average value and the mean absolute deviation (in parenthesis) over the data set considered.

$<$0.5) however, the distance measures are poorly correlated with each other, with most correlation coefficients in the range 0.5 to 0.7. Both results confirm that when two structures are very close to each other, different distance measures quantify their differences in a similar manner. When the two structures however are very different, different distance measures will focus on different geometric differences, leading to differences in their behaviors. We observe however one exception in Table 6.2, in that RMSD and MT clearly remains correlated (0.80) even for the diverse subgroup of TSA with TM $<$ 0.5. The reason for this exception is unclear.

The CASP 10 Stage_1 and Stage_2 test sets usually include longer proteins than the other sets considered here, with decoys that are far from their native counterparts. In the Stage_1 sets there are very few decoys per target (by construction, see Methods above) and relatively large average mean deviations of the distance measures. For the Stage_2 test sets there are more decoys per target; these decoys however are usually very similar to each other, leading to very low mean absolute deviations for the GDT-TS* and Q* distance measures, and consequently to low correlations and R scores between the

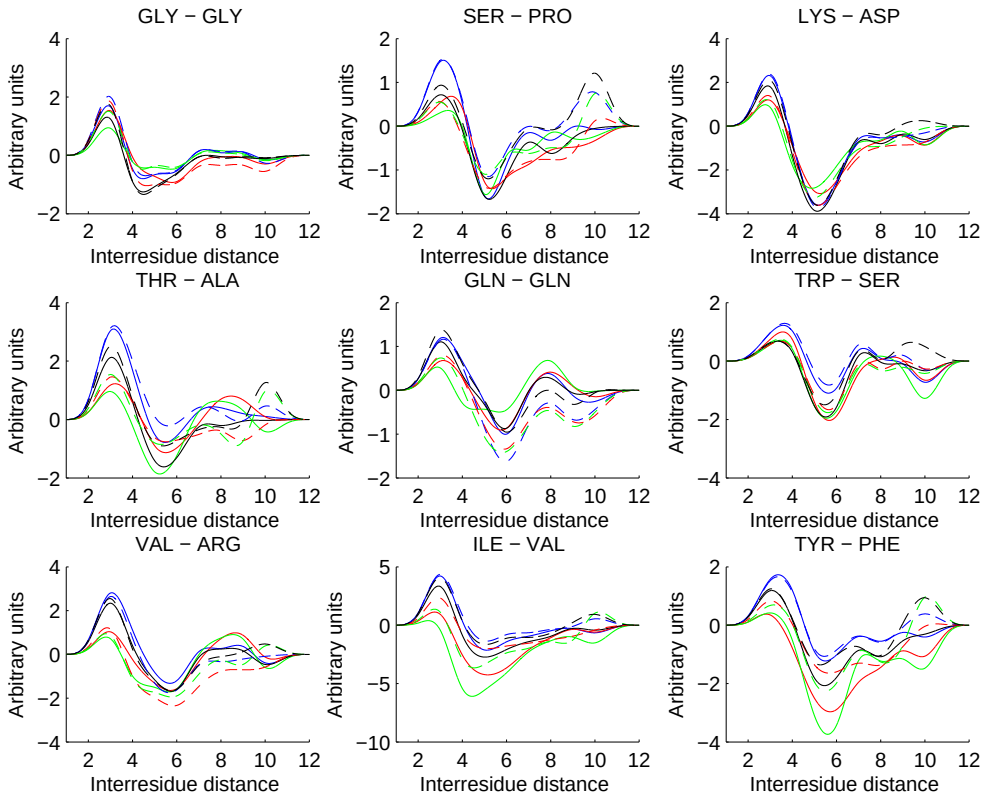


Figure 6.1: Showing nine different types of residue pair interactions for our single model method PPD (continuous lines) and our consensus method PPE (dotted lines) when trained on RMSD (blue), MT(red), GDT-TS(green) and Q(black).

measures. As an example, the correlation between RMSD and GDT-TS* for the Stage_2 decoy sets is only 0.51 and their non symmetric R scores are R(RMSD,GDT-TS*)=0.71 and R(GDT-TS*,RMSD)=0.73, respectively. These low values are good indicators of significant differences between their ranking of the decoys included in CASP10 Stage_2 test sets.

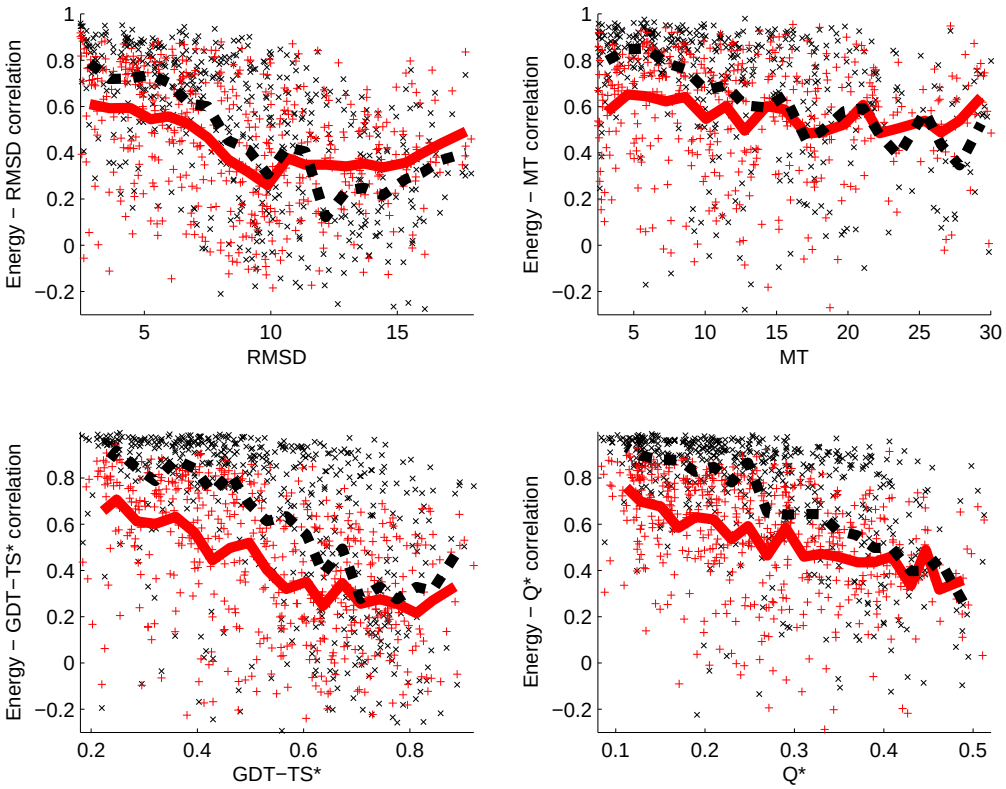


Figure 6.2: **Energy-distance correlations as a function of the quality of the decoy set.** For each decoy set in Titan-HRD, CASP-HRD, and TSA (a total of 797 sets), we plot the correlation Corr(E, d_1) as a function of the mean value of d_1 over the decoy set, where E is either the PPD energy (red, plus sign +) or the PPE energy (black, cross sign x) trained on the set Titan-HRD with the distance measure d_1, and d_1 is one of the fourth distance measures considered, namely RMSD (panel A), MT (panel B), GDT-TS* (panel C), and Q* (panel D). The corresponding running means computed over 20 equidistant intervals for PPD (red, solid line) and PPE (black, dashed line) are shown. Clearly, the quality of the correlation energy-distance decreases as the diversity of the decoy set increases.

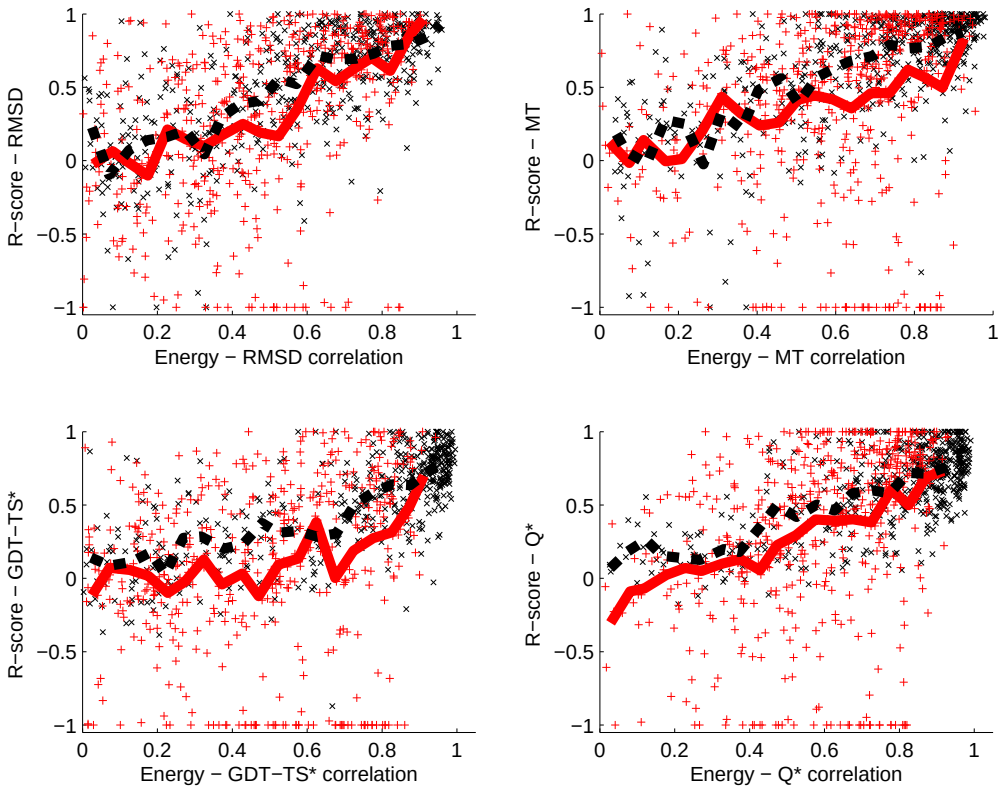


Figure 6.3: **R scores versus Energy-distance correlations.** For each decoy set in Titan-HRD, CASP-HRD, and TSA, we plot the R score $R(d_1, E)$ as a function of the correlation coefficient $Corr(d_1, E)$, where E is either the PPD energy (red, plus sign +) or the PPE energy (black, cross sign x) trained on the set Titan-HRD with the distance measure d_1, and d_1 is one of the fourth distance measures considered, namely RMSD (panel A), MT (panel B), GDT-TS* (panel C), and Q* (panel D). The corresponding running means computed over 20 equidistant intervals for PPD (red, solid line) and PPE (black, dashed line) are shown. Note that $R(d_1, E)$ compares the best decoy picked based on the energy value E with the decoy closest to the native structure according to the distance measure d_1. There is a clear correlation between these two values for all four distance measures.

6.3.2 Training knowledge-based potentials with different distance measures.

We have derived two new smooth knowledge-based residue pair potentials, PPD and PPE. Both potentials are based on distances between the $C\alpha$ atoms of the protein structure of interest. For each of the 210 types of amino acid pairs, the two potentials are written as a weighted sum of smooth spline functions, whose weights are optimized so that the total energy of a protein model resembles the distance between the model and a reference structure (usually taken to be the native structure), as described by equation 6.1. The two potentials differ however on which pairs of residues are taken into account. While PPD includes all pairs of residues from the protein structure P considered, PPE only include those pairs whose inter $C\alpha$ distance is consistently below a cutoff value in an ensemble of protein models similar to P. The idea behind PPE, derived from Eickholt et al. [105], is that the various models in the ensemble contain complementary information which can be pooled together to build a contact map of consistent residue-residue contacts that are more likely to be informative. Our interest here is to assess the influence of the distance measure used to train the two potentials. We have trained PPD and PPE on the Titan-HRD* training set with the four distance measures introduced above separately, and tested the corresponding four versions of the potentials against the Titan-HRD, CASP-HRD, and TSA test sets in their abilities to mimic any of the four distance measures. All parameters describing the amino acid pair spline potentials are listed in the file SupportingMaterialData. The encoding used and the spline basis used is described in the file SupportingMaterialReadme. Both files are in the supporting information.

Figure 6.1 shows some examples of the b-spline expanded pair potentials. As expected, the pair potentials are repulsive for short inter-residue distances and have a first minimum between 4 Å and 6 Å and this preferred distance relatively independent of the training metric. For longer pair distances it is seen that most PPD pair potentials have a local minimum around 10Å whereas the PPE pair potentials tend to have a local maximum at this distance. One plausible explanation is that as PPE does not identify new contacts for these large distances; it may then set higher energy values for remote decoys. The exact placement of the minimum as well as the depth of the potential differs for the different pair potentials. While these differences may seem small, they add up when we sum over all the interactions.

We computed both the correlations between energy and the distance measure, and the R scores that compare the best decoys picked based on energy with the decoys closest to their corresponding native structures. Results are given in Table 6.4 for the correlation coefficients, Table 6.5 for the R scores, and in Figures 6.2 and 6.3 for a comparison of these scores. We draw from these tables and figures the four main conclusions described below.

First, we find that both potentials PPD and PPE perform very well on the Titan-HRD test set, for all distance measures used for training and testing the potential. The corresponding mean correlation coefficients (averaged over all decoys sets in Titan-HRD) are usually above 0.8, indicating that the energy functions order the decoys in the same manner as the distance measures. In parallel, the R scores are also high, with most values well above 0.65, indicating that the decoys with the lowest energies are usually among the decoys that are close to the corresponding native structures. We should note however that PPD and PPE were trained on Titan-HRD*. While Titan-HRD and Titan-

HRD* are different (see Methods), they both contain decoys that were generated with the same principles, with the significant constraint that they maintain the hydrophobic cores of the corresponding native structures. The exceptional performance of PPD and PPE may therefore not be surprising in light of this comment. Indeed, as we test these potentials on different decoy sets with more diverse populations of decoys, we observe a decrease in performance that follows the increase in diversity (in the order Titan-HRD - TSA (TM > 0.5) - CASP-HRD - TSA (TM < 0.5). This decrease in performance is illustrated in Figure 6.2.

Second, the ensemble potential PPE performs better than the single structure potential PPD, again for all the distance measures used to train and test the potentials. The differences between the two potentials are large for the high resolution decoys sets in Titan-HRD and TSA (TM > 0.5), but become statistically insignificant for very diverse decoy sets such as those in TSA (TM < 0.5). We believe that these differences illustrate the power of generating consensus information from an ensemble. In PPE, we only consider those contacts there are consistently below a given distance cutoff in the whole decoy set to which the protein of interest belongs. This initial filtering is clearly an advantage for Titan-HRD, as it will select the contacts in the hydrophobic cores which are native, and will ignore the contacts that fluctuate significantly due to the sampling procedure used to generate the decoys. It remains an advantage for high quality decoy but becomes less pertinent for highly diverse decoys.

Third, the performances of the two potentials PPD and PPE depend on the choice of the distance used in the training step. For example, the correlations between PPE and any of the four distance measures increase on average by 0.09 when it is trained on MT instead of RMSD (Table 6.4). Similar differences are observed for the R scores between PPE and the four distance measures (Table 6.5). More generally, it is best to train the potentials on a distance measure that is directly based on intrinsic inter-residue distances, such as MT that follows the elastic network of the protein of interest, or Q* that counts the number of contacts that fall below a given distance cutoff, than on a distance measure based on extrinsic Euclidean distances, such as RMSD. Interestingly, we find that GDT-TS* behaves more like the intrinsic distance measures MT and Q* than RMSD, even though it is also based on extrinsic distances. The reason for this discrepancy is unclear.

Finally, we observe that the ability of an energy function to pick a "good" decoy (i.e. with native-like characteristics) is contingent to how well this energy function correlates with a distance measure between decoys and native structure. This is illustrated in Figure 6.2. This observation validates the approach of sculpting (training) a potential to mimic a distance measure.

Table 6.4: Energy-distance correlations

		PPD				PPE				RAPDF [a]	GOAP[b]	AMBER [c]
		Training distance d_1 [d]				Training distance d_1 [d]						
Decoy set	Test Distance d_2 [d]	RMSD	MT	GDT-TS*	Q*	RMSD	MT	GDT-TS*	Q*			
Titan-HRD	RMSD	0.77 (0.05) [e]	0.82 (0.04)	0.81 (0.05)	0.79 (0.04)	0.81 (0.05)	0.88 (0.03)	0.85 (0.03)	0.82 (0.04)	0.5 (0.14)	0.64(0.11)	0.01 (0.02)
	MT	0.82 (0.04)	0.89 (0.03)	0.87 (0.03)	0.87 (0.03)	0.86 (0.04)	0.95 (0.01)	0.92 (0.02)	0.89 (0.02)	0.47 (0.16)	0.63(0.11)	0.01 (0.02)
	GDT-TS*	0.83 (0.06)	0.91 (0.02)	0.91 (0.02)	0.9 (0.03)	0.86 (0.04)	0.93 (0.02)	0.95 (0.01)	0.92 (0.02)	0.43 (0.18)	0.63(0.12)	0.001 (0.02)
	Q*	0.82 (0.05)	0.92 (0.2)	0.92 (0.02)	0.93 (0.02)	0.87 (0.03)	0.95 (0.01)	0.97 (0.01)	0.95 (0.01)	0.37 (0.22)	0.57(0.13)	0.002 (0.02)
CASP-HRD	RMSD	0.32 (0.16)	0.31 (0.16)	0.31 (0.16)	0.26 (0.16)	0.42 (0.16)	0.51 (0.17)	0.51 (0.18)	0.45 (0.17)	0.31(0.15)	0.3 (0.15)	0.01 (0)
	MT	0.45 (0.11)	0.49 (0.11)	0.49 (0.12)	0.43 (0.12)	0.55 (0.13)	0.69 (0.11)	0.68 (0.13)	0.61 (0.13)	0.37 (0.14)	0.41(0.13)	0.02 (0)
	GDT-TS*	0.38 (0.14)	0.39 (0.13)	0.39 (0.13)	0.33 (0.14)	0.51 (0.15)	0.6 (0.14)	0.65 (0.14)	0.58 (0.17)	0.39 (0.14)	0.43(0.11)	0.02 (0)
	Q*	0.46 (0.12)	0.6(0.11)	0.64 (0.1)	0.57 (0.1)	0.54 (0.12)	0.69 (0.12)	0.75 (0.12)	0.71 (0.1)	0.32 (0.15)	0.42(0.12)	0.02 (0)
CASP10-stage1	RMSD	0.44 (0.22)	0.53 (0.18)	0.53 (0.18)	0.48 (0.2)	0.5 (0.22)	0.54 (0.22)	0.52 (0.21)	0.5 (0.21)	0.18(0.24)	0.32 (0.26)	-0.03 (0)
	MT	0.47 (0.19)	0.61(0.17)	0.62 (0.17)	0.55 (0.18)	0.56 (0.16)	0.63 (0.13)	0.61 (0.13)	0.57 (0.13)	0.13 (0.22	0.34(0.24)	-0.06 (0)
	GDT-TS*	0.4 (0.21)	0.49 (0.23)	0.51 (0.2)	0.43 (0.21)	0.57 (0.21)	0.63 (0.16)	0.63 (0.16)	0.59 (0.2)	0.22 (0.28)	0.4(0.2)	-0.05 (0)
	Q*	0.51 (0.22)	0.63(0.16)	0.63 (0.16)	0.56 (0.18)	0.68 (0.12)	0.75 (0.06)	0.75 (0.07)	0.72 (0.1)	0.24 (0.3)	0.41(0.2)	-0.05 (0)
CASP10-stage2	RMSD	0.33 (0.18)	0.34 (0.2)	0.34 (0.2)	0.31 (0.2)	0.31 (0.18)	0.37 (0.19)	0.34 (0.16)	0.3 (0.21)	0.15(0.13)	0.2 (0.14)	-0.005 (0)
	MT	0.42 (0.16)	0.49 (0.16)	0.49 (0.15)	0.45 (0.14)	0.4 (0.16)	0.5 (0.19)	0.48 (0.15)	0.42 (0.16)	0.19 (0.14)	0.29(0.14)	0.003 (0)
	GDT-TS*	0.31 (0.15)	0.29 (0.14)	0.29 (0.13)	0.25 (0.14)	0.37 (0.16)	0.42 (0.19)	0.44 (0.18)	0.38 (0.18)	0.29 (0.14)	0.37(0.17)	0.007 (0)
	Q*	0.45 (0.22)	0.56(0.14)	0.58 (0.12)	0.52 (0.15)	0.51 (0.18)	0.62 (0.17)	0.66 (0.14)	0.62 (0.15)	0.28(0.13)	0.41(0.13)	-0.006 (0)
TSA	RMSD	0.62 (0.12)	0.62 (0.13)	0.63 (0.13)	0.59 (0.14)	0.74 (0.09)	0.8 (0.07)	0.78 (0.08)	0.73 (0.09)	0.5 (0.14)	0.58(0.13)	0.02 (0.01)
	MT	0.65 (0.11)	0.69 (0.1)	0.7 (0.1)	0.65 (0.12)	0.75 (0.08)	0.83 (0.06)	0.8 (0.06)	0.74 (0.07)	0.5 (0.16)	0.58(0.12)	0.03 (0.01)
TM-score > 0.5	GDT-TS*	0.6 (0.15)	0.59 (0.14)	0.6 (0.13)	0.54 (0.16)	0.78 (0.06)	0.85 (0.04)	0.84 (0.04)	0.79 (0.06)	0.61 (0.11)	0.7(0.1)	0.03 (0.01)
	Q*	0.69 (0.11)	0.71 (0.09)	0.72 (0.1)	0.68 (0.1)	0.87 (0.04)	0.94 (0.02)	0.93 (0.02)	0.9 (0.03)	0.57 (0.13)	0.67(0.11)	0.03 (0.01)
TSA	RMSD	0.3 (0.16)	0.34 (0.18)	0.34 (0.18)	0.32 (0.18)	0.29 (0.23)	0.36 (0.27)	0.34 (0.27)	0.29 (0.23)	0.16 (0.13)	0.25(0.15)	0 (0.01)
	MT	0.38 (0.14)	0.47 (0.15)	0.47 (0.15)	0.45 (0.17)	0.34 (0.23)	0.45 (0.24)	0.41 (0.23)	0.35 (0.22)	0.19 (0.14)	0.29(0.13)	-0.003(0.01)
TM-score < 0.5	GDT-TS*	0.27 (0.19)	0.27 (0.2)	0.28 (0.19)	0.24 (0.18)	0.36 (0.29)	0.44 (0.33)	0.42 (0.32)	0.36 (0.29)	0.26 (0.16)	0.33(0.19)	0.004 (0.02)
	Q*	0.41 (0.17)	0.47 (0.16)	0.46 (0.17)	0.45 (0.15)	0.53 (0.19)	0.63 (0.18)	0.61 (0.18)	0.57 (0.19)	0.23 (0.17)	0.3(0.19)	-0.004 (0.02)

[a] All-atom statistical distance-based potential [116].
[b] All-atom orientation-dependent statistical potential [85].
[c] The semi-empirical physical potential AMBER99SB-ILDN[117].
[d] PPD and PPE have been trained on the distance measure d_1 and tested against the distance measure d_2.
[e] Average value, and mean absolute deviation (in parenthesis) over the data set.

Table 6.5: Energy-distance Rvalues

		PPD				PPE				RAPDF [a]	GOAP [b]	AMBER [c]
		Training distance d_1 [d]				Training distance d_1 [d]						
Decoy set	Test Distance d_2 [d]	RMSD	MT	GDT-TS*	Q*	RMSD	MT	GDT-TS*	Q*			
Titan-HRD	RMSD	0.57 (0.17)	0.52 (0.17)	0.51 (0.19)	0.48 (0.2)	0.61 (0.14)	0.63 (0.17)	0.62 (0.14)	0.61 (0.15)	0.43 (0.18)	0.56 (0.15)	0.26 (0.47)
	MT	0.73 (0.1)	0.71 (0.11)	0.69 (0.13)	0.69 (0.15)	0.76 (0.11)	0.8 (0.11)	0.79 (0.12)	0.8 (0.1)	0.5 (0.18)	0.6(0.17)	0.23 (0.53)
	GDT-TS*	0.78 (0.08)	0.74 (0.09)	0.74 (0.08)	0.73 (0.09)	0.82 (0.07)	0.86 (0.06)	0.86 (0.07)	0.86 (0.07)	0.52 (0.22)	0.67(0.12)	0.18 (0.58)
	Q*	0.73 (0.11)	0.76 (0.12)	0.73 (0.1)	0.77 (0.1)	0.77 (0.09)	0.84 (0.09)	0.85 (0.09)	0.86 (0.08)	0.37 (0.19)	0.53(0.17)	0.16 (0.51)
CASP-HRD	RMSD	0.19 (0.31)	0.00 (0.42)	-0.04 (0.48)	0.03 (0.4)	0.27 (0.3)	0.33 (0.31)	0.34 (0.27)	0.3 (0.31)	0.14 (0.37)	0.22(0.37)	-0.11 (0.34)
	MT	0.31 (0.26)	0.24 (0.3)	0.14 (0.31)	0.16 (0.36)	0.38 (0.24)	0.43 (0.22)	0.43 (0.21)	0.38 (0.25)	0.24 (0.4)	0.46(0.32)	-0.09 (0.47)
	GDT-TS*	0.14 (0.3)	-0.09 (0.4)	-0.08 (0.42)	-0.06 (0.43)	0.28 (0.23)	0.31 (0.25)	0.32 (0.24)	0.28 (0.24)	0.12 (0.44)	0.34(0.33)	-0.22 (0.38)
	Q*	0.27 (0.25)	0.35 (0.33)	0.34 (0.32)	0.33 (0.31)	0.28 (0.2)	0.39 (0.24)	0.4 (0.25)	0.41 (0.26)	0.13 (0.4)	0.43(0.26)	-0.17(0.42)
CASP10-stage1	RMSD	0.55 (0.23)	0.53 (0.3)	0.57(0.26)	0.48 (0.39)	0.5 (0.29)	0.52 (0.26)	0.53 (0.26)	0.51 (0.27)	0.32 (0.34)	0.44(0.26)	0.13 (0.6)
	MT	0.69(0.1)	0.7(0.12)	0.72(0.12)	0.64(0.14)	0.58(0.15)	0.62(0.12)	0.63(0.13)	0.6(0.15)	0.37(0.28)	0.52(0.2)	0.16(0.6)
	GDT-TS*	0.52(0.32)	0.47(0.39)	0.52(0.37)	0.42(0.43)	0.43(0.4)	0.46(0.37)	0.51(0.33)	0.46(0.36)	0.27(0.44)	0.53(0.22)	0.15(0.37)
	Q*	0.6(0.24)	0.64(0.15)	0.67(0.14)	0.57(0.19)	0.57(0.22)	0.6(0.18)	0.63(0.17)	0.61(0.18)	0.32(0.38)	0.47(0.31)	0.19(0.5)
CASP10-stage2	RMSD	0.38(0.29)	0.23(0.36)	0.29(0.3)	0.26(0.35)	0.36(0.31)	0.35(0.32)	0.32(0.32)	0.35(0.34)	0.29(0.28)	0.39(0.29)	0.11(0.42)
	MT	0.55(0.23)	0.46(0.31)	0.5(0.29)	0.49(0.33)	0.45(0.32)	0.47(0.32)	0.47(0.32)	0.48(0.3)	0.44(0.32)	0.52(0.24)	0.17(0.45)
	GDT-TS*	0.23(0.35)	0.11(0.33)	0.14(0.36)	0.14(0.34)	0.25(0.32)	0.23(0.28)	0.29(0.32)	0.25(0.32)	0.23(0.3)	0.39(0.3)	0.01(0.27)
	Q*	0.45(0.32)	0.46(0.31)	0.51(0.29)	0.48(0.3)	0.44(0.3)	0.47(0.24)	0.51(0.27)	0.53(0.27)	0.33(0.28)	0.41(0.27)	0.13(0.43)
TSA	RMSD	0.47 (0.24)	0.22 (0.41)	0.21 (0.41)	0.22 (0.41)	0.69 (0.14)	0.69(0.13)	0.69 (0.13)	0.68 (0.14)	0.44 (0.25)	0.59(0.19)	0.24 (0.38)
	MT	0.62 (0.14)	0.4 (0.24)	0.39 (0.27)	0.41 (0.24)	0.8 (0.08)	0.81 (0.07)	0.82 (0.08)	0.8 (0.08)	0.54 (0.18)	0.74(0.1)	0.32 (0.33)
TM-score > 0.5	GDT-TS*	0.29 (0.28)	0.09 (0.43)	0.08 (0.47)	0.09 (0.45)	0.58 (0.16)	0.6 (0.16)	0.61 (0.16)	0.57 (0.18)	0.37 (0.31)	0.56(0.41)	0.09 (0.26)
	Q*	0.49 (0.19)	0.32 (0.3)	0.3 (0.33)	0.33 (0.28)	0.68 (0.14)	0.72 (0.14)	0.74 (0.13)	0.74 (0.13)	0.38 (0.29)	0.59(0.43)	0.14 (0.2)
TSA	RMSD	0.16 (0.35)	0.19 (0.3)	0.19 (0.34)	0.16 (0.32)	0.26 (0.35)	0.33 (0.37)	0.32 (0.36)	0.29 (0.35)	0.19 (0.35)	0.27(0.4)	0.04 (0.41)
	MT	0.27 (0.34)	0.38 (0.3)	0.39 (0.33)	0.37 (0.29)	0.36 (0.33)	0.44 (0.31)	0.42 (0.32)	0.41 (0.31)	0.28 (0.34)	0.4(0.33)	0.04 (0.46)
TM-score < 0.5	GDT-TS*	0.07 (0.3)	0.07 (0.28)	0.1 (0.3)	0.06 (0.29)	0.26 (0.3)	0.32 (0.3)	0.32 (0.3)	0.29 (0.3)	0.19 (0.3)	0.28(0.36)	0.05 (0.3)
	Q*	0.18 (0.35)	0.28 (0.34)	0.29 (0.35)	0.3 (0.31)	0.42 (0.27)	0.5 (0.27)	0.49 (0.29)	0.49 (0.27)	0.19 (0.29)	0.31(0.32)	0.01 (0.31)

[a] All-atom statistical distance-based potential [116].
[b] All-atom orientation-dependent statistical potential [85].
[c] The semi-empirical physical potential AMBER99SB-ILDN[117].
[c] PPD and PPE have been trained on the distance measure d_1 and tested against the distance measure d_2.
[d] Average value, and mean absolute deviation (in parenthesis) over the data set.

6.3.3 Comparison with other energy functions

We have compared the two energy functions PPD and PPE with two well established all-atom statistical potentials RAPDF[116] and GOAP[85] and with a semi-empirical physical potential, AMBER99SB-ILDN [117], for all decoy sets in Titan-HRD, CASP-HRD, and TSA. Results for correlations between energy and distance measures and for R scores are given in Tables 6.4 and 6.5, respectively.

As intuitively expected, the performances of AMBER99SB-ILDN are very poor. This is most likely an artifact due to the presence of a few steric clashes in the decoys, and not a reflection of the quality of this potential. While it would be possible to improve this performance by applying an initial energy minimization on all decoys, this result by itself highlights that such a physical potential cannot be used directly to order a set of decoys, unless some pre-processing is applied.

RAPDF is a knowledge-based statistical potential that is based on a direct conversion of the distributions of inter-atomic distances observed in native protein structures into energy values that are then used to assess how native-like a model is [116]. It is not based on any information from existing decoy sets, and it is not trained to mimic some differences between decoys and native structures. It is therefore not surprising that it does not perform as well as PPD and PPE, especially on the Titan-HRD as both PPD and PPE were trained on decoys resembling those included in this data set.

GOAP is an all-atom orientation-dependent knowledge-based statistical potential that includes a distance-based term and an angle-dependent contribution [85]. The distance-based term is an all-atom statistical potential that is based on the reference state that was introduced with the DFIRE potential [118]. The angle dependent component of GOAP is based on the geometric orientation of local planes. GOAP is found to perform significantly better than RAPDF on all datasets tested in this study. This is not a surprise, as GOAP includes much more information than RAPDF due to its angle term. We find however that GOAP performs only marginally better than PPD and worse than PPE. This illustrates the benefit of training a potential on a decoy set. PPD and PPE are only Ca based potentials; they have been trained however to mimic distances between non-native models and native structures of proteins.

The performances of RAPDF and GOAP depend on the distance measure used for testing. We observe that they are particularly good when the statistical potentials are tested on GDT-TS*, reflecting the differences between these distance measures (see Table 6.2 and 6.3).

6.3.4 Performance in the CASP 10 quality assessment category

As part of the CASP experiment, state-of-the-art methods for protein structure assessment are judged on their ability to evaluate the quality of the predictions submitted as models for the targets considered in that specific experiment: this is the quality assessment category (QA). In 2012 as part of CASP10, 37 groups participated [95]. They were asked to evaluate the quality of sets of predictions (decoys) in two rounds designated as Stage_1 (20 decoys with a large variation in quality as measured by GDT-TS) and Stage_2 (150 decoys with homogeneous quality as measured by GDT-TS). The main reason for providing a small number of decoys in Stage_1 was to allow for judging assessment methods that rely on a single model independently from methods that rely on an ensemble of decoys (consensus methods), that would be tested extensively with the

Stage_2 decoy sets. The three main conclusions drawn from these experiments were [95]: 1) The performances of the single model methods are usually worse than the the performances of consensus methods, 2) The Stage_2 sets are usually more difficult to rank than the Stage_1 sets, and 3) No methods were able to consistently pick the best decoy in an ensemble. The results for the participating groups can be seen in Figure 6.2 (average correlation) and Figure 6.3 (ability to pick the best decoy) in [95]. We note that the single model method GOAP used in this study differs from the quasi-single model method GOAPQA used in CASP10QA. For the latter, the TM-score [115] to the top 5 ranked models is used as a measure of model quality.

The CASP 10 datasets have average native-decoy RMSDs of 11-13Å. These differences are significantly larger than the 2.4Å RMSD found in our training sets (see Table 6.1). Our analyses of the performances of PPD (single model) and PPE (ensemble of decoys) on the other datasets considered in this study have shown that for decoys that are far from their native counterparts, the two methods perform similarly, and in fact poorly (see top left panel of Figure 6.2 and Table Table 6.4). We observe the same behavior when PPD and PPE are applied on the CASP10 datasets (Tables 6.4 and 6.5). Similarly we expect and indeed find that the ensemble method PPE is ineffective in ranking the decoys of the CASP10 datasets when its performance is measured against the MT distance measure, and shows some prospects when its performance is measured against the GDT-TS* and Q* distance measures. The energy-GDT-TS correlations of 0.51(0.63) and 0.29(0.44) for PPD(resp. PPE) on Stage_1 and Stage_2 respectively are amongst the lowest reported for single model(resp. ensemble) methods in CASP10QA [95]. The low energy-distance correlations reported usually leads to a bad pick for the best decoy, see Figure 6.3. It is therefore surprising that the average ΔGDT-TS* of 0.07 between the GDT-TS*-closest decoy and the lowest energy decoy picked by PPE on the CASP10 Stage_2 data sets places PPE in the middle of the CASP10 participating methods (see [95] Figure 2(A)).

The results for PPD, PPE, AMBER99SB-ILDN, RAPDF and GOAP on CASP 10 stages 1 and 2 are given in Tables 4 - 6 where PPD and PPE were trained and tested on the same distance measure. Clearly, GOAP has a better performance than PPD when GDT-TS* is chosen as a measure of distance. It is however noteworthy that PPD performs better than GOAP when measured by RMSD and MT instead. It is encouraging that the distance dependent C-alpha potential, PPD, as a single model method has a performance that is comparable to the state-of-the-art orientation-dependent all-atom potential, GOAP. We find that PPD is good at selecting a decoy that is close to the native structure (Table 6).

6.4 Concluding Remarks

The recent literature on generating knowledge-based potentials for protein structure modeling makes no secrets of their limitations and problems. Knowledge-based potentials are energy functions derived primarily from databases of protein structures and sequences. They can be divided into two classes. Potentials from the first class are based on a direct conversion of the distributions of some geometric properties observed in native protein structures into energy values, while potentials from the second class are trained to mimic quantitatively the geometric differences between incorrectly folded models (also called decoys) and native structures. Both potentials are designed to as-

sess how native-like a model structure is. There is no consensus however on which geometric property should be considered, on how to convert a statistical distribution into an energy for the first class, and on how energy and geometry should be related in the second class.

In this paper, we focused on the relationship between energy and geometry when training knowledge-based potentials from the second class. We assumed that the difference between the energy of a decoy and the energy of its corresponding native structure must be linearly related to the distance between the decoy and the native structure. We trained two distance-based Cα potentials accordingly, one based on all inter-residue distances (PPD), while the other had the set of all these distances filtered to reflect consistency in an ensemble of decoys (PPE). Compared to other methods that follow the same approach however, we did not assume that the distance between a decoy and the native structure is the traditional RMSD. Instead, we tested four different distance measures, two based on extrinsic geometry (RMSD and GTD-TS*), and two based on intrinsic geometry (Q* and MT). We found that it is usually better to train the potentials using the latter type of distances.

We have found that both PPD and PPE perform extremely well on the high resolution decoy set Titan-HRD, with correlation coefficients between energy and distance usually well above 0.8. PPE always performs better than PPD on this set, emphasizing the benefits of capturing consistent information in an ensemble. While we trust the general trends highlighted by these results, we tone down the importance of In extensive testing on available decoy sets and models from the Critical Assessment of protheir exceptional character as they may only reflect the specificity of the Titan-HRD data set. tein Structure Prediction (CASP) experiments we find that PPD yields better energy-distance correlations than one of the state of the art single model potentials, GOAP [85]. We note however that the sophisticated distance-based and orientation-based statistical potential GOAP is better at picking the best decoys and has a better though comparable performance for fixed energy-distance correlation. It should be noted that PPD and PPE are Cα-based, while GOAP is an all-atom potential. We believe that this demonstrates that a very efficient training of a simple distance-based pair potential can generate a very effective measure for assessing protein structure models.

There is still room for improvement in training knowledge-based potentials. We limited our study to pairwise potentials; we will test different geometric properties of protein structures in future studies. We plan to include the potentials described here into a structure minimization package, to assess their performances in improving non-native protein structure models.

6.5 Acknowledgments

The authors want to thank the anonymous reviewers for constructive criticism and careful reading of the first version of this manuscript.

Table 6.6: Assessing the best decoys selected by energy functions on different decoy datasets

		Best	PPD	PPE	RAPDF [a]	AMBER [b]	GOAP [c]
Titan-HRD	RMSD	1.1(0.21)[d]	1.7(0.29)	1.6(0.27)	1.9(0.4)	2.1(0.55)	1.7(0.3)
	MT	0.75(0.22)	1.4(0.38)	1.2(0.38)	1.8(0.62)	2.3(0.89)	1.6(0.54)
	GDT-TS	0.94(0.02)	0.89(0.03)	0.92(0.03)	0.85(0.05)	0.8(0.09)	0.88(0.03)
	Q	0.94(0.01)	0.92(0.02)	0.93(0.02)	0.88(0.03)	0.86(0.04)	0.89(0.03)
4-state	RMSD	1.1(0.1)	3.8(0.44)	2.2(0.21)	2.1(0.22)	3.6(1.5)	1.6(0.24)
	MT	0.9(0.33)	5.8(2.1)	1.2(0.31)	2.6(0.52)	6.2(3.4)	1.5(0.38)
	GDT-TS	0.91(0.03)	0.55(0.06)	0.86(0.08)	0.8(0.04)	0.67(0.1)	0.86(0.04)
	Q	0.94(0.02)	0.75(0.03)	0.92(0.02)	0.87(0.04)	0.79(0.1)	0.9(0.02)
fisa	RMSD	3.7(0.76)	5.7(0.78)	6.5(1.4)	4.4(0.72)	8.5(1.5)	4.5(0.45)
	MT	3.8(1.5)	7.9(3.7)	5.5(2)	5.4(2.5)	10(4.2)	4.9(1.9)
	GDT-TS	0.65(0.07)	0.51(0.14)	0.54(0.08)	0.6(0.06)	0.46(0.08)	0.59(0.06)
	Q	0.82(0.02)	0.79(0.03)	0.78(0.03)	0.78(0.02)	0.73(0.02)	0.79(0.03)
fisa CASP3	RMSD	6(2)	12(1.6)	12(2.4)	12(4)	12(1)	11(1.6)
	MT	8.7(4.2)	21(7.3)	17(8.2)	23(4.5)	19(2.5)	18(7.7)
	GDT-TS	0.47(0.12)	0.32(0.01)	0.34(0.02)	0.32(0.02)	0.29(0.01)	0.33(0.04)
	Q	0.76(0.06)	0.72(0.04)	0.72(0.04)	0.68(0.04)	0.67(0.07)	0.69(0.06)
hg Structal	RMSD	1.9(0.5)	2.6(1)	2.5(0.56)	2.2(0.5)	3.3(0.71)	2.4(0.6)
	MT	1.8(0.3)	2.5(0.61)	3(0.28)	2.4(0.35)	3.7(0.8)	2.7(0.28)
	GDT-TS	0.86(0.06)	0.82(0.14)	0.85(0.07)	0.84(0.07)	0.77(0.08)	0.84(0.08)
	Q	0.93(0.03)	0.92(0.03)	0.92(0.03)	0.92(0.04)	0.89(0.03)	0.92(0.04)
lmds	RMSD	5.7(0.33)	9.9(0.72)	9.8(0.89)	9.8(0.92)	10(0.61)	10(0.65)
	MT	8(0.78)	14(3.7)	17(5.5)	16(2.5)	19(1.6)	19(4.5)
	GDT-TS	0.45(0.04)	0.29(0.04)	0.32(0.05)	0.31(0.05)	0.28(0.03)	0.3(0.03)
	Q	0.74(0.02)	0.67(0.05)	0.67(0.04)	0.65(0.04)	0.63(0.03)	0.63(0.05)
lattice ssfit	RMSD	3.8(0.46)	7.6(1.3)	7.4(1.6)	7.7(1.9)	8(2.6)	8.5(1.2)
	MT	5.2(2.2)	9.8(4.5)	10(5.1)	11(4.8)	12(6.4)	12(5)
	GDT-TS	0.62(0.06)	0.45(0.07)	0.48(0.07)	0.49(0.12)	0.45(0.07)	0.44(0.04)
	Q	0.8(0.06)	0.74(0.07)	0.75(0.04)	0.74(0.05)	0.72(0.07)	0.72(0.06)
CASP5	RMSD	6.7(2.9)	13(6.2)	11(6)	10(5.2)	11(6.1)	10(5.2)
	MT	8.5(4.2)	20(11)	18(7.7)	20(12)	22(11)	20(9.9)
	GDT-TS	0.58(0.19)	0.36(0.17)	0.48(0.24)	0.44(0.19)	0.46(0.21)	0.5(0.23)
	Q	0.82(0.09)	0.72(0.13)	0.77(0.09)	0.7(0.1)2	0.72(0.13)	0.75(0.1)
CASP6	RMSD	4.8(1.5)	10(5.1)	11(4.5)	9.7(5.1)	12(5.9)	8(3.1)
	MT	5.4(1.9)	23(11)	18(4.3)	19(5.7)	24(15)	12(4)
	GDT-TS	0.64(0.14)	0.33(0.14)	0.52(0.18)	0.49(0.27)	0.38(0.17)	0.54(0.17)
	Q	0.85(0.06)	0.7(0.07)	0.79(0.08)	0.75(0.11)	0.68(0.11)	0.79(0.09)
CASP7	RMSD	4.5(1.8)	8.8(4.9)	7.1(3.1)	7.9(3.9)	11(5.1)	7.8(3.4)
	MT	3.8(1.6)	9.5(3.8)	6.6(2.8)	10(4.3)	18(9.1)	8.3(3.5)
	GDT-TS	0.66(0.13)	0.49(0.21)	0.56(0.14)	0.56(0.17)	0.43(0.21)	0.58(0.13)
	Q	0.88(0.04)	0.81(0.08)	0.85(0.06)	0.81(0.06)	0.74(0.08)	0.82(0.06)
CASP8	RMSD	4.1(1.3)	7.4(2.7)	6.4(1.8)	9.8(5.5)	9.7(5.2)	7.5(3.1)
	MT	3.2(1.3)	10(4.1)	6.1(2.2)	14(6.4)	15(8)	8.7(2.7)
	GDT-TS	0.7(0.1)	0.53(0.17)	0.63(0.13)	0.51(0.22)	0.47(0.19)	0.61(0.16)
	Q	0.89(0.04)	0.81(0.07)	0.85(0.06)	0.79(0.09)	0.75(0.1)	0.83(0.07)
CASP9	RMSD	4.8(1.4)	9.7(4.9)	7.5(2.6)	9.4(4.7)	9.8(4.9)	8.2(3.3)
	MT	3.7(1.3)	14(7.8)	7.3(2.6)	12(3.9)	13(5.2)	8.5(2.6)
	GDT-TS	0.68(0.1)	0.4(0.15)	0.6(0.12)	0.52(0.19)	0.51(0.21)	0.57(0.15)
	Q	0.88(0.04)	0.75(0.12)	0.85(0.04)	0.8(0.09)	0.79(0.09)	0.83(0.07)
TASSER Set II	RMSD	3.2(1)	5.6(1.9)	5.2(1.4)	5.2(1.6)	6.4(2.1)	5.4(1.9)
	MT	3.7(1.2)	7.1(2.3)	6.6(2.5)	6.7(2.2)	11(5.4)	6.8(2.2)
	GDT-TS	0.69(0.09)	0.57(0.12)	0.59(0.12)	0.59(0.1)	0.52(0.13)	0.59(0.12)
	Q	0.85(0.05)	0.81(0.06)	0.82(0.06)	0.8(0.05)	0.75(0.08)	0.79(0.05)
Rosetta-All	RMSD	6.4(1.3)	11(2.1)	11(2.2)	12(3.2)	16(4.7)	11(2.6)
	MT	12(4.7)	22(8.9)	26(7.2)	25(9.9)	47(14)	24(7.5)
	GDT-TS	0.41(0.06)	0.29(0.04)	0.29(0.04)	0.28(0.04)	0.25(0.04)	0.28(0.04)
	Q	0.72(0.06)	0.64(0.07)	0.64(0.07)	0.62(0.07)	0.59(0.07)	0.62(0.07)
Rosetta-Baker	RMSD	4.7(2.2)	7.5(3.4)	8.4(4.3)	7.6(4.1)	8.2(2.7)	6.9(3.6)
	MT	6.9(3)	13(7.5)	15(9.2)	13(7.8)	13(6.8)	11(5.4)
	GDT-TS	0.6(0.21)	0.47(0.15)	0.46(0.13)	0.48(0.15)	0.46(0.13)	0.5(0.18)
	Q	0.84(0.08)	0.77(0.1)	0.77(0.11)	0.77(0.01)	0.76(0.07)	0.79(0.09)
Rosetta-Tsai	RMSD	2.8(0.8)	6.9(2.8)	5(1.4)	7.3(1.8)	5.7(2)	6.2(2.1)
	MT	3(1.1)	9.3(4.9)	5.5(2.2)	10(5.8)	8.3(3)	7.1(2.3)
	GDT-TS	0.72(0.08)	0.47(0.08)	0.59(0.09)	0.45(0.06)	0.54(0.09)	0.52(0.12)
	Q	0.86(0.05)	0.77(0.08)	0.81(0.08)	0.74(0.07)	0.77(0.07)	0.77(0.06)
CASP-HRD	RMSD	2(0.56)	2.6(0.73)	2.6(0.71)	2.6(0.76)	2.8(0.7)	2.6(0.74)
	MT	1.1(0.5)	2(0.8)	1.7(0.61)	2(0.83)	2.3(0.11)	1.7(0.79)
	GDT-TS	0.83(0.07)	0.75(0.06)	0.78(0.07)	0.77(0.07)	0.75(0.06)	0.78(0.07)
	Q	0.93(0.03)	0.9(0.03)	0.91(0.03)	0.89(0.04)	0.88(0.04)	0.91(0.03)
CASP10-stage1 [e]	RMSD	4.7(1.2)	6.6(2.3)	7.1(2.4)	8.2(3.2)	12(4.7)	7.3(2.4)
	MT	4(2)	6.6(2)	8(1.6)	10(3.5)	20(3.4)	7.9(2.2)
	GDT-TS	0.71(0.1)	0.62(0.13)	0.63(0.15)	0.57(0.16)	0.55(0.19)	0.63(0.14)
	Q	0.88(0.04)	0.84(0.04)	0.85(0.04)	0.82(0.06)	0.8(0.06)	0.84(0.06)
CASP10-stage2 [e]	RMSD	4(1.1)	5.9(1.7)	5.9(1.6)	6.7(1.9)	9(2.2)	6.2(1.6)
	MT	3.1(1)	5.4(1.6)	5.8(1.6)	6.6(1.8)	14(1.9)	5.4(1.5)
	GDT-TS	0.73(0.09)	0.64(0.14)	0.66(0.11)	0.65(0.12)	0.65(0.11)	0.67(0.11)
	Q	0.89(0.04)	0.86(0.03)	0.87(0.04)	0.86(0.05)	0.85(0.04)	0.86(0.04)

[a] All-atom statistical distance-based potential [116].
[b] The semi-empirical physical potential AMBER99SB-ILDN[117].
[c] All-atom orientation-dependent statistical potential [85].
[d] Average value, and mean absolute deviation (in parenthesis) over the data set.
[e] Only ensembles who contains a decoy with a $GDT - TS > 0.4$ are included. Compare with Figure 2 in [95]

Chapter 7

Protein structure refinement by optimization

M. Carlsen, P. Røgen, Protein structure refinement by optimization, *Proteins: Structure, Function and Bioinformatics*, accepted, 2015.

Abstract. Knowledge-based protein potentials are simplified potentials designed to improve the quality of protein models which is important as more accurate models are more useful for biological and pharmaceutical studies. Consequently, knowledge-based potentials often are designed to be efficient in ordering a given set of deformed structures denoted decoys according to how close they are to the relevant native protein structure, but this does not necessarily imply that energy minimization of this potential will bring the decoys closer to the native structure. In this study, we introduce an iterative strategy to improve the convergence of decoy structures. It works by adding energy optimized decoys to the pool of decoys used to construct the next and improved knowledge-based potential. We demonstrate that this strategy results in significantly improved decoy convergence on Titan high resolution decoys and refinement targets from Critical Assessment of protein Structure Prediction competitions. Our potential is formulated in Cartesian coordinates and has a fixed backbone potential to restricts motions to be close to those of a dihedral model, a fixed hydrogen-bonding potential and a variable coarse grained carbon alpha potential consisting of a pair potential and a novel solvent potential that are b-spline based as we use explicit gradient and Hessian for efficient energy optimization.

Keywords: Protein structure refinement, knowledge-based potentials, iterative methods, optimization, funnel sculpting

7.1 Introduction

The three dimensional structure of proteins is predicted with methods such as comparative modelling, if structural information of similar proteins is known and de novo methods if not[2]. The predicted structures do not always achieve sufficiently high accuracy for them to be used for applications such as drug design, drug screening, ligand docking or molecular replacement[4]. With the purpose of improving the accuracy of protein structures, every second year several groups participate in the model refinement category in the Critical Assessment of protein Structure Prediction (CASP)[6]. The number of participating research groups has increased remarkedly from CASP 7 to CASP 10 and, as a consequence, a number of new methods has been developed. Despite a growing interest, the improvements in the refinement targets have been small and the methods have not shown the consistency necessary for them to be used in practise[6]. A consistent method to improve the accuracy of protein structures is, therefore, desirable and this work is a step in that direction.

Two different strategies have been developed to refine targets: potential energy minimization (PEM)[26, 27, 28] and molecular dynamics (MD)[29, 30, 31]. Potential energy minimization is a deterministic technique originally used in chemistry to remove high energies in a molecule. The method, however, is meaningful to use as a refinement method when the energy landscape of a potential function is funnel-shaped - at least in a neighborhood of the region where the structure is refined. The minimization algorithm then tumbles down towards the lowest possible state. On the other hand, molecular dynamics whether deterministic or stochastic is a method that can overcome the obstacles in the landscape due to the kinetic energy of the molecule. It, however, is computationally more demanding than PEM.

The main problem in protein structure refinement is that the structure tends to move the model away from the native structure[26]. The reason for this misbehavior may be errors in the PEM-potentials or MD-potentials. The MD-potentials are capable of folding fast-folding proteins[119] but the energy landscape for these proteins is particularly simple and a more complex energy landscape with valleys, hills and several transition states and local minima is expected for the CASP refinement targets. Nonetheless, a MD-based method was the best of all methods to refine CASP 10 refinement targets which is remarkable as it is the first time a MD-based method wins this category[6].

The starting point of this work is metric training[12] of a knowledge-based potential where all parameters defining the potential are given by optimizing the linear relation between decoy-native energy difference and decoy-native distance. In Ref. [120] it is investigated which notion of distance that is most efficient for metric training of pair potentials. Here we use an anharmonic elastic network model and found it very efficient for this propose yielding average energy-distance correlations of 0.89 for near native decoys. However, even if the constructed potential is efficient in ordering a given set of decoys, there is no guaranty that energy minimization brings the decoys closer to the native structure. Often a low energy structure close to, and seemingly in an arbitrary direction from, the initial decoy is found. In this work, we therefore follow an iterative strategy for improving the decoys convergence toward native structures during energy minimization. Given a metrically trained potential we minimize a set of decoys and add the resulting structures back into the pool of decoys. The new decoys will have lower energy compared to their distance to the native structure than the old decoys. In a renewed metric training it is thus favorable to raise the energies in the new decoys

relatively to the old decoys. Hereby, the worst escape directions of the previous potential have been assigned a relatively higher energy and energy minimization in the next potential will follow a different and hopefully better path. The idea is similar to the strategy used in Ref. [121, 57], but here we focus on near native structural refinement and the metric training allows us to work on many (here 924 non-homologous) proteins and all their decoys simultaneously. Another example of optimizing a potential by alternating between using it for energy minimization and re-optimizing the potential is [122, 123]. Here one fixed sequence is first threaded against a library of structures and the best hits are then relaxed by stochastic energy minimization in the potential. The hereby generated structures are used to optimize the potential through optimizing the z-score as function of the relative strengths of otherwise fixed energy terms. References to optimization of a potential based on a fixed set of structures are given in [12]. Corresponding to de novo modeling we perform a test on decoy sets of non-homologous proteins and of non-homologous CASP refinement targets. We also perform an experiment to see how convergent refinement can become when the potential is specialised for one protein.

Our potential is formulated in Cartesian coordinates and consists of standard terms being a fixed backbone potential that restricts motions to be close to those of a dihedral model, a fixed hydrogen-bonding potential and a variable coarse grained carbon alpha part consisting of a pair potential and a novel solvent potential. The variable part of the potential is b-spline based as we use explicit gradient and Hessian for efficient energy optimization.

7.2 Methods

7.2.1 A backbone model of a protein

We use a backbone model of a protein where each amino acid contains the atoms N, H, C_α, C and O (except for proline which only has four backbone atoms). This is sufficient to sustain the backbone structure and the hydrogen bonds in the protein. The backbone is sustained by restricting the motions mainly to the phi/psi torsion angles by adding harmonic restraints to the bond lengths, bond angles and rotation about the peptide bonds. A hydrogen bond is found in a given native structure using the DSSP definition and sustained by adding harmonic restraints to the bond length $H - O$. We refer to the sum of the backbone potential and the hydrogen potential as the local potential E_{L}. The equilibrium bond lengths, bond angles, torsion angles and hydrogen bonds are always taken from the initial structure and the spring constants for E_{L} are fixed at 1.

The global potential which we designate as E_{G} only depends on the C_α atoms and consists of a pair potential E_{Pair} and two solvent potentials E_{S5} and E_{S9}. For each of the 210 amino acid pairs the pair potential is defined as a cubic uniform b-spline of the distance between the C_α atoms. The pair potential tends smoothly to zero at distance 9Å and is parametrized using the 8 basis functions shown on Fig. 7.1.

Solvent effects are often quantified using the solvent accessible surface area [38, 39] but current models also include a volume based term[124]. Solvation effects were modeled as an example based on calculations of solvent-accessible surface area and solvent-excluded volumes for the residues of a protein represented as a union of balls of radius 5Å centered at each C_α atom[12]. Here we make a smooth alternative that reuse the b-spline functions calculated for the pair potential. It is based on the

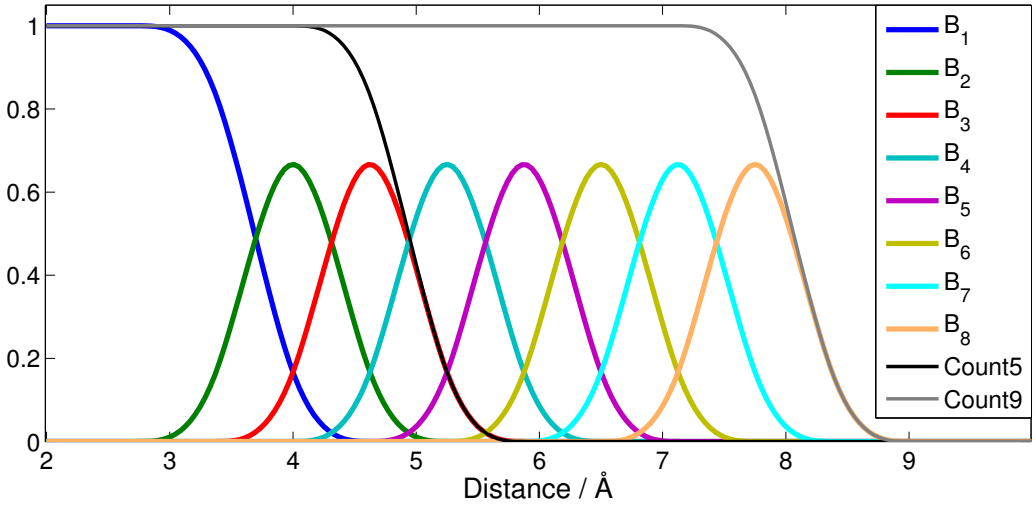


Figure 7.1: The spline basis $B_1, \ldots, B_8$ for the pair potential and the partition of unity $\mathrm{Count5}$ and $\mathrm{Count9}$ used for the solvent potential.

observation that the accessible surface area and solvent-excluded volume both correlate to the number of neighbors. For the 2590 cases where an amino acid occurs only ones in a native structure used in the Titan-HRD, see Data sets, we find the accessible surface area and solvent-excluded volume calculated in Ref. [12] highly correlated (0.97) and the smooth counting of the number of neighbors, $non9$, defined below has a clear correlation of -0.87 and -0.89 to the two quantities. We also include a smooth counting of the number of near neighbors $non5$ as it is relatively independent of $non9$.

The smooth partition of unity $\mathrm{Count5}\left(\mathrm{pair}\right) = \sum_k B_k\left(d_{\mathrm{pair}}\right)$ is one for $d_{\mathrm{pair}} \leq 4$ and tends smoothly to zero for $d_{\mathrm{pair}} = 5.8$ (See Fig. 7.1). For a given residue p we define its number of neighbours at distance 5 as $\mathrm{non5_p} = \sum_{q \neq p} \mathrm{Count5}\left(\mathrm{pair_{p,q}}\right)$ and finally its associated potential as

$$E_{S5_p} = \sum_{m=1}^{8} a_m \tilde{B}_m\left(\mathrm{non5}_p\right),$$

where the a_m's depend on the amino acid type and $\tilde{B}_m$ is a uniform cubic b-spline between -0.5 and 5. E_{S9} is defined similairly but here $\mathrm{non9}_p$ is defined using $\mathrm{Count9}\left(\mathrm{pair}\right) = \sum_k B_k\left(d_{\mathrm{pair}}\right)$ which is one for $d_{\mathrm{pair}} \leq 7$ and tends smoothly to zero for $d_{\mathrm{pair}} = 9$ (See Fig. 7.1) and $\tilde{B}_m$ is a uniform b-spline between -3.25 and 27. The solvent potential thus has 320 parameters. The global potential, E_G, thus has 2000 parameters in total to be determined.

7.2.2 Metrics

The decoy-native energy gab for the pair potential and the two solvent potentials are estimated with three measures. Let subindex i refer to the i-th native structure and indices i, j refer to the j-th decoy of the i-th native structure. A pair of amino acids in the native structure are said to be in contact if their distance $d_i \leq 8$. Denoting the distance between the amino acid pair in decoy j by $d_{i,j}$, the energy gab for the pair

potential E_{Pair} is estimated with the anharmonic potential[103] defined by

$$NT_{i,j} = \frac{1}{N_{\text{c}}} \sum_{d_i \leq 8\mathring{A}} \left(e^{-(d_{i,j}-d_i)/2} + (d_{i,j} - d_i)/2 - 1 \right), \tag{7.1}$$

where the sum is taken over all contacts and N_c denotes the number of contacts. NT has the same Hessian as the usual elastic network potential $\sum_{d_i \leq 8\mathring{A}} (d_{i,j} - d_i)^2$ but has the desired property to grow linearly for large values of $d_{i,j} - d_i$.

The energy gab for E_{S5} is estimated in the following way: Equivalent to the definition above we define the total number of neighbors for a given residue p within an sphere of radius 5Å as $\text{non}5_p$. The metric is then defined as

$$Q5_{i,j} = \sum_{\text{p}} \left(\text{non}5_{i,p} - \text{non}5_{i,j,p} \right)^2 \tag{7.2}$$

where the sum is taken over all amino acids. We define $Q9_{i,j}$ in the same manner but using $\text{non}9$ which is closely related to accessible surface area. $Q9_{i,j}$ therefore models changes in solvation interaction. Our final metric is a combination of $NT_{i,j}$, measuring changes to native amino acid pair contacts and $Q5_{i,j}$ and $Q9_{i,j}$ measuring collective changes to native amino acid neighborhoods including solvation. The chosen combination $NT_{i,j} + \alpha_i Q9_{i,j} + \beta_i Q5_{i,j}$ gives equal numerical influence to the three measures by fixing the constants α_i and β_i by a linear fit, $NT_{i,j} \approx \alpha_i \cdot Q9_{i,j}$ and $NT_{i,j} \approx \beta_i \cdot Q9_{i,j}$.

7.2.3 Parameter optimization

The parameters for the local potential are fixed at arbitrary high values whereas the parameters for the global potential are determined in the parameter optimization. The parameter optimization introduced in [12] ideally wants the decoy-native energy gap to equal the decoy-native distance. As this is not possible it instead minimizes the sum of squared errors between the native-decoy energy gab and a metric $NT_{i,j} + \alpha_i Q9_{i,j} + \beta_i Q5_{i,j}$:

$$f(\boldsymbol{X}) = \sum_{i,j} \| E_{G_{i,j}}(\boldsymbol{X}) - E_{G_i}(\boldsymbol{X}) - NT_{i,j} - \alpha_i \cdot Q9_{i,j} - \beta_i \cdot Q5_{i,j} \|^2 + \gamma \cdot \|\boldsymbol{X}\|^2, \tag{7.3}$$

where $\boldsymbol{X}$ is a vector with 2000 variables and the last term is added since we are looking for a Tikhonov regularized solution which ensures that the quadratic matrix has full rank. We fix γ at 10^{-2}.

In Ref. [12] where the energy gab was estimated by the root-mean-square deviation (RMSD) the funnel around each native structure was allowed to have its own slope. By the normalization of the energy gab introduced here this is not necessary reducing the number of parameters quite substantially by the number of native structures.

7.2.4 Structural optimization

The algorithm which we use for PEM is a modified Newton method whose descent directions combines the gradient and directions of negative curvature of the Hessian[58, 59]. The non-convex Hessian based method searches for a local minimum that satisfies the first and second order optimality conditions. This means that the gradient is vanishing

Table 7.1

Training set	x	y	z	w	$x \times y \times z \times w$	l[Å]
Single	1	10	20	1	200	1
Titan-HRD	100	10	20	10	200000	0.5

and the Hessian is positive semidefinite of the potential when evaluated at the optimum. We used the method described in [125] to calculate the gradient and Hessian of the local and global potential. One of the advantages of using this method is that vectorization techniques can be used in the implementation. Vectorization is a technique in Matlab where vector and matrix operations are used instead of loops. One of the advantages of using the expressions derived in this work is that the block matrices are expanded in terms of operators which are outer products and thus vectorizable. The function accumarray is used to build the gradient and Hessian which has a sparse matrix as output option. This function takes six arguments as input of which the first three and the last here are relevant (see Matlab documentation for details). In the first two arguments the index and the matching values are given as inputs (a vector with 1 or 2 columns for the gradient and Hessian respectively which specifies where in the matrix the values are going to be placed). In the third argument, the size of the matrix is specified and in the last argument we may specify whether the output matrix is sparse or dense. In this way, it is fast to place the block matrices at their right position in the Hessian. The principle is the same whether we implement the derivatives of bond lengths, bond angles or torsion angles. Overall, the gradient and the Hessian of a molecular potential can be calculated without having to use a single loop. This leads to fast evaluation times in high-level languages. The implementation was done in Matlab (The Mathworks, Inc., Natick, MA).

7.2.5 Improving decoy-convergence

Our iterative method to improve decoy-convergence shifts between a parameter optimization of the potential and a structural optimization of decoys. In the first round, we sculpt the surface using the optimization procedure described above and an initial training set (see Data sets below). Next, we minimize a set of x random decoy ensembles containing y random decoys until the minimization has moved l Angstrom away from the initial decoys. From the minimization runs we select z new decoys which have been chosen as uniformly as possible. The $x \times y \times z$ new decoys in total are finally added to the training set with a weight factor w. At subsequent rounds, $x \times y \times z$ new decoys are produced at each round. The basic idea behind the iterative procedure is to raise the energy of the decoys that have moved away from the native structure. The parameters of the initial training sets can be found in Table 1.

7.2.6 Data sets

The training set used is Titan-HRD. Titan-HRD is generated using the torsion angle dynamics program DYANA41 subject to distance constraints to preserve the hydrophobic core of a protein[47, 107]. It is assumed that the hydrophobic core includes all residues within a β strand as well as all hydrophobic residues within an α-helix. The decoys

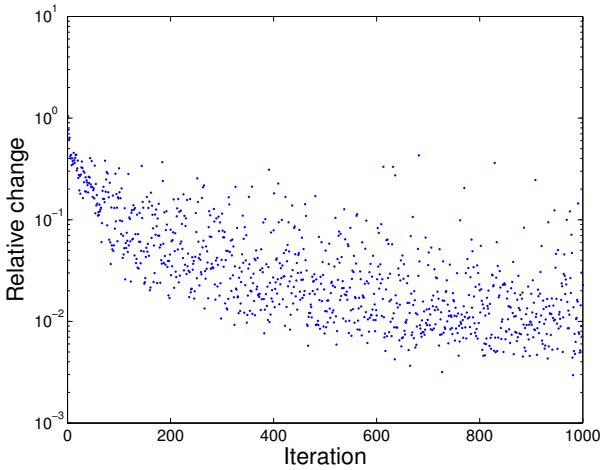


Figure 7.2: The size of the relative change $\|\boldsymbol{X}_i - \boldsymbol{X}_{i+1}\|_2 / \|\boldsymbol{X}_i\|_2$ between the i-th and $i+1$-th parameter sets are shown for each iteration in the first experiment.

are generated by adding constrains to the distance between the hydrophobic and hydrophilic amino acid and relaxing these thus controlling the nearness of the decoys.

We use two different sets from the Titan-HRD set. The first set consists exclusively of one native structure (CATH code 1-10-10-10) and 1066 near-native decoys. We divide this set into a training set of 966 decoys and a independent test set of 100 decoys. The second set includes 1066 non-homologous native structures in total and on average 946 near-native decoys pr. native structure. From this set we use 924 native-decoy ensembles as a training set and 142 native protein and decoy ensembles as an independent test set. Note that as the Titan-HRD native structures are non-homologous this is also the case for the training and test set. Finally, we test our potentials on a collection of 37 refinement targets picked from the CASP 7 to 10 refinement experiments. The Titan-HRD predates the CASP targets and no significant sequence similarity is present between the two sets. We denote these test set by $\mathrm{T}_{\mathrm{Single}}$, $\mathrm{T}_{\mathrm{Titan\text{-}HRD}}$ and $\mathrm{T}_{\mathrm{CASP}}$, respectively.

7.3 Results

A consistent force field is a force field that improves the resolution of decoys and most importantly does no harm to it. The purpose of this work is to show that the iterative procedure that we have developed generates a force field that is more consistent than the raw force field. We perform two experiments in total.

The first experiment is meant to give an upper bound on the obtainable performance of structural refinement through energy optimization and we thus only sculpt the energy landscape in a region about one native structure. We first train the raw potential and the decoy-convergence improved potential on a single native-decoy ensemble and test the potential on a set of decoys that have been generated in the same manner as the training set. Fig. 7.2 shows the convergence of the parameters for this experiment. Apparently, the convergence is slow and the fluctuations of the norm large. We decided to stop the algorithm after 1000 iterations. The result of this experiment is shown in Fig. 7.3. We see that the force field based on the iterative method outmatch the raw

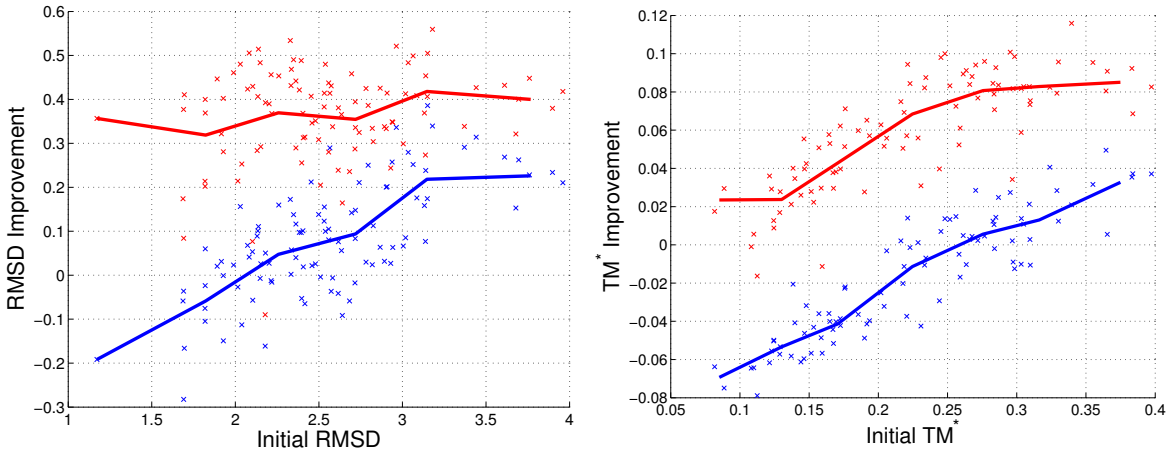
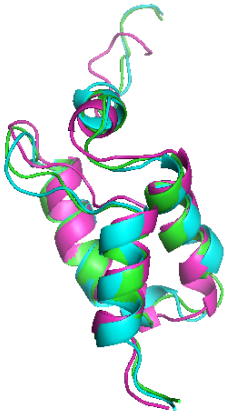


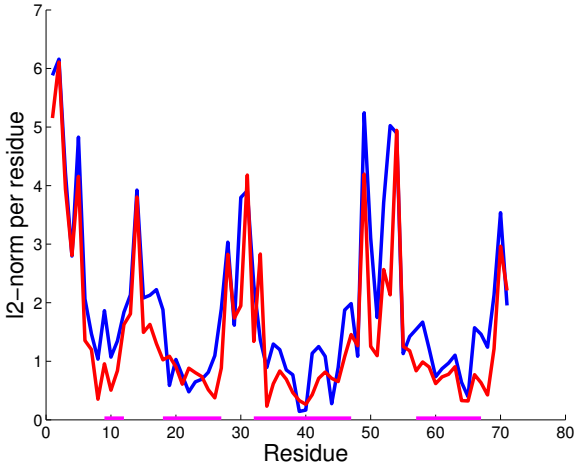
Figure 7.3: Data for the first experiment on the T_{Single} test set. Left: Showing the RMSD improvements (defined as the the initial minus the final RMSD to the native structure) as a function of initial RMSD. Right: Showing the TM* improvements (defined as the the initial minus the final TM* to the native structure) as a function of initial TM*. The results were obtained by energy minimization when allowing up to 1Å RMSD derivation from the initial decoy for the raw potential (blue) and the decoy-convergence improved potential (red). The two solid lines are means with a bin width of 0.5.

force field and achieves considerable improvements in RMSD and TM*[1]. The decoy-convergence improved force field is very consistent since the quality of almost all of the decoys has been improved. This is by no means the case for the raw force field where about $1/4$ of the decoys have larger RMSD to the native structure after the structural optimization. Fig. 7.4a shows the RMSD alignment of the native, decoy and energy refined decoy and Fig. 7.4b carbon alpha displacements for each residue in this alignment. The main conformational differences in the decoy are found in the loop regions but the most consistent improvements obtained by the energy minimization are related to the arrangement of the four helices and the loop residues close to the helices. Next, we consider a more difficult experimental setup. We train and test the potentials on a large set of non-homologous protein ensembles. This experiment is more difficult than the first experiment as we require that the knowledge-based potential is capable of sculpting the energy landscape around a large set of proteins all non-homologous to the training proteins. We stopped the iterative decoy-convergence improvement procedure after 500 iterations in total involving 500.000 PEMs of length 0.5 Angstrom. It is expected that the parameters of the potential are only slowly converging as only half of the original decoys have been subject to PEM as illustrated in Fig. 7.5. The initial potential is quite efficient in ordering the decoys as the average correlation between the potential and the native-decoy training distance $NT_{i,j} + \alpha_i \cdot Q9_{i,j} + \beta_i \cdot Q5_{i,j}$ on the test set $T_{Titan\text{-}HRD}$ is 0.78 and compatible with the corresponding number of 0.79 reported in both [12, 47]. Even though our iterative process adds new decoys to the training, this high energy-distance correlation is not disturbed on the original decoys. In fact we find the average energy-distance correlation to improve to 0.87 after the iterative process,

[1]TM* is defined as 1-TM where TM is introduced in Ref. [115]. The TM-score has the advantage when compared to RMSD that the score is less affected by chain length and outliers.



(a) Native model of 1eijA (magenta) shown together with the RMSD aligned initial model (blue; 2.4 Angstrom to native structure) and refined model (green; 2 Angstrom to native structure).



(b) For each residue the carbon alpha displacement between the native model and the initial model (blue) and the final model (red) respectively. The residues in secondary structures have been coloured magenta.

Figure 7.4

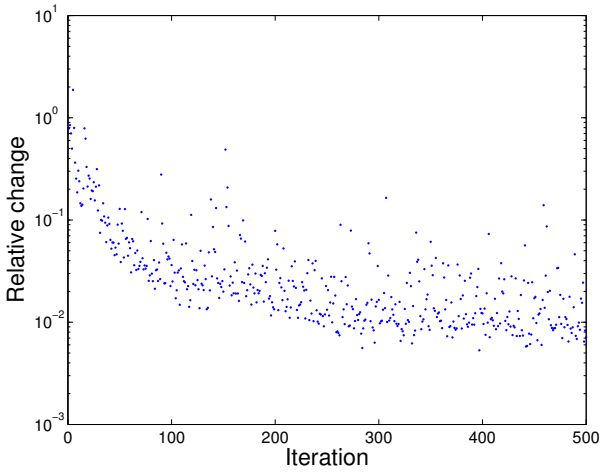


Figure 7.5: The size of the relative change $\|\boldsymbol{X}_i - \boldsymbol{X}_{i+1}\|_2 / \|\boldsymbol{X}_i\|_2$ between the i-th and $i+1$-th parameter sets are shown for each iteration in the second experiment.

which is close to the 0.89 reported in [120] when training and testing a pair potential on the elastic network energy NT. All these high correlations are reported on the Titan High Resolution Decoys and are expected to decrease if calculated on decoys with lower resolution[120]. The results for the second experiment are illustrated in Fig. 7.6. We observe that the RMSD improvements and TM* improvements of the initial decoys have a significant dependence on the initial RMSD and TM* and the convergence of PEM is almost uniformly improved by the iterative procedure. In fact, all targets with RMSD greater than 2 Angstrom and TM* greater than 0.1 are most likely to be improved. Hence both potentials are more consistent the greater the initial RMSD or TM* is. We also note a consistency of the performance seen on the Titan HRD as the results are

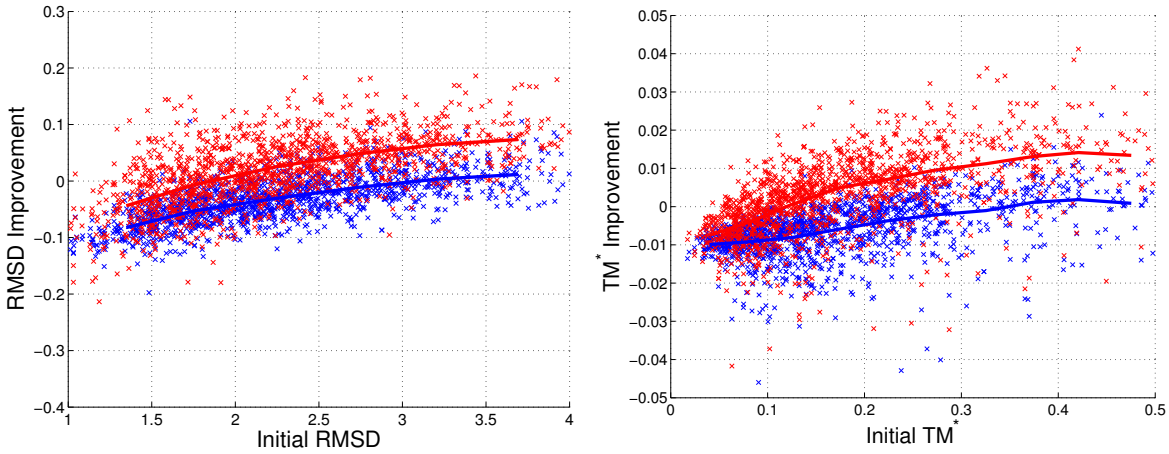


Figure 7.6: Data for the second experiment on the $T_{\text{Titan-HRD}}$ test set. Showing the RMSD improvements (left) and TM* improvements (right) obtained by energy minimization when allowing up to 0.5Å RMSD deviation from the initial decoy for the raw potential (blue) and the decoy-convergence improved potential (red) as a function of initial RMSD (left) and initial TM* (right). The two solid lines are means with a bin width of 0.5.

almost the same if we perform the test on the Titan-HRD training set (data not shown) instead of the non-homologous test set.

We also tested the two potentials from the second experiment on 37 CASP refinement targets. We did this to test the raw potential and the decoy-convergence improved potential on a difficult and realistic test set. The set is difficult as it has been generated by many different prediction methods that our potentials have no knowledge about. Also each model is chosen as the best model provided by a number of prediction servers[6] and they are thereby better than the current prediction performance of the field. The results shown in Table 2 are however encouraging for the targets that have a large RMSD or TM*. The previous test showed that both the raw potential and the decoy-convergence improved potential are more consistent the greater the initial RMSD or TM* of a decoy is. We find the same tendency for the CASP refinement targets. Overall we improve 50% of the targets which is an acceptable result compared to the 31% reported in CASP10 [6].

For the first and second experiment that we have performed we set a limit at 1 Angstrom and 0.5 Angstrom, respectively, as the maximal displacement of a decoy. The setting of this limit is of course a subjective choice. We therefore found it to be an interesting investigation whether the experiments would have turned out differently if we had set a different limit. The results for the experiments with a different limit can be found in the supporting material. Here, Fig. S1a(S1b) and S1c(S1d) correspond to Fig. 7.3 for a limit at 0.5 Angstrom and 4 Angstrom for RMSD (the TM-score) and Fig. S2a (S2b) and Table S1 correspond to Fig. 7.6 and Table 2 for a limit at 1 Angstrom for RMSD (the TM-score). The result of the first experiment confirms our choice of 1 Angstrom as a reasonable limit. It would have been a very conservative choice had we chosen a limit at 0.5 Angstrom instead. On the contrary, when we let the rutine run until it reaches 4 Angstrom or until it converges (often it is seen that the rutine converges before 4 Angstrom) we find that there is are much greater probability of doing harm to

the structure. The choice of 1 Angstrom is thus a compromise between avoiding doing harm to the structure and having displacements of a reasonable size. The limit of 1 Angstrom for the second experiment is however questionable. The method becomes much more consistent if we reduce this limit to 0.5 which thus was our prefered choice.

Overall, for all of the experiments we performed, we find that that decoy-convergence improved potential outmatches the raw potential in both experiments. This result is very satisfactory in that we have shown that the iterative method works in different experimental setups.

7.4 Conclusions

We introduce a smooth knowledge based protein potential with explicit gradient and Hessian that both are needed for computationally efficient energy minimization and run energy minimizations that each is terminated after 0.5Å RMSD deformation from the initial structure. Our iterative strategy to improve decoy-convergence of the potential works consistently for all native-model distances represented. Models at least 3Å RMSD from the native structure are generally brought closer to their native structure by energy minimization, whereas near native models generally are degraded by energy minimization. The library of decoy structures for both test and training is of non-homologous proteins. Hence, our experimental setup corresponds to de novo protein modeling which typically provides models in the range $4 - 8$Å form the native structure where our method seems applicable. We get a peak in the direction of comparative modelling by optimizing structural refinement around a single native structure (as e.g. done in [122, 123, 126]). In this case we get decoy convergence for all decoys independent of initial native-decoy RMSD.

We next allow the energy minimization to deform models up to 1Å RMSD from the initial structure and find energy minimization seemingly capable of performing larger structural changes as a local minimum almost never is found after 1Å deformation. For 1Å deformations the iterative strategy also consistently improves the performance of energy minimization in the potential; but even the improved potential generally degrades the decoys. Hence, even if an energy minimization at first improves the model it generally degrades it when allowing larger conformational changes. This effect appears less pronounced on the CASP refinement targets where targets improved by 0.5Å RMSD energy minimizations generally also are improved by 1.0Å RMSD energy minimizations. We expect the structural drift of longer energy minimizations and the convergence limit of 3Å RMSD for shorter energy minimizations both mainly to be caused by the functional form and coarse graining of the energy function. I.e., we expect that the refinement limit may only be pushed significantly closer to the native structure by expanding the functional form of the potential or by narrowing the range of its application; e.g., to that of homology modeling where a larger number of sequence similar, and thus likely also structurally similar, structures are known. Both directions are subject for further investigations.

Our results on the CASP refinement targets with respect to RMSD were neither worse nor better than most of the groups participating in the CASP 10 refinement category. As reported in Ref. [6] on the evaluation of the CASP 10 refinement targets the majority of the groups fail to improve the quality of targets. As such it is not surprising that our

methods also fail to improve several of the models. Our main result is therefore the significant increase in decoy convergence we observe after using the iterative method.

Finally, the iterative procedure to improve decoy-convergence can be used for any potential designed by, or partially by, its performance on decoys. The linear parameter dependence of the potential used here is not strictly necessary, but it gives a computationally efficient shaping of the potential when combined with the least squares formulation of metric training. Similarly, the procedure does not require deterministic energy minimization which may be exchanged with, e.g., a stochastic method[57, 126]. It is thus our hope that other research groups will use a similar strategy to improve the decoy-convergence of their knowledge-based potentials.

7.5 Acknowledgements

The authors want to thank Mathias Stolpe for discussions and careful reading of the manuscript. The authors also want to thank the anonymous reviewers for their constructive criticism of the first version of this manuscript.

Target	RMSD$_i$	TM$_i^*$	ΔRMSD$_1$	ΔTM$_1^*$	ΔRMSD$_2$	ΔTM$_2^*$
TR432	1.65	0.08	-0.08	-0.01	-0.15	-0.02
TR453	1.40	0.11	-0.08	-0.02	-0.22	-0.02
TR454	3.24	0.20	0.02	-0.00	0.03	-0.00
TR461	1.63	0.07	-0.07	-0.01	-0.07	-0.01
TR464	2.94	0.27	-0.07	-0.01	-0.03	0.00
TR469	2.18	0.26	-0.07	-0.02	-0.35	-0.06
TR476	6.77	0.57	0.06	0.00	0.24	0.01
TR488	2.11	0.12	-0.04	-0.01	-0.08	-0.02
TR517	4.65	0.23	-0.04	-0.01	0.00	-0.01
TR530	1.99	0.15	-0.07	-0.02	-0.04	-0.02
TR557	4.07	0.26	-0.03	-0.00	0.03	0.00
TR567	3.44	0.16	-0.00	-0.01	0.09	-0.00
TR568	6.15	0.44	-0.03	-0.01	-0.07	-0.02
TR574	3.58	0.35	-0.07	-0.01	0.05	-0.00
TR592	1.26	0.08	-0.02	-0.01	-0.04	-0.01
TR594	1.82	0.10	-0.05	-0.01	-0.05	-0.01
TR606	4.85	0.27	-0.03	-0.01	0.01	0.00
TR622	7.47	0.28	0.03	-0.00	0.16	0.00
TR624	5.19	0.49	-0.04	-0.00	0.06	-0.00
TR644	2.71	0.14	-0.01	-0.01	-0.04	-0.01
TR661	2.74	0.11	-0.02	-0.01	0.03	-0.00
TR662	1.92	0.19	-0.09	-0.02	-0.16	-0.02
TR663	3.37	0.23	0.01	-0.00	0.10	0.00
TR671	7.72	0.49	0.02	-0.00	0.06	-0.00
TR674	3.44	0.13	0.03	-0.01	0.08	-0.00
TR679	3.95	0.18	-0.02	-0.01	-0.01	-0.00
TR696	3.52	0.27	-0.04	-0.01	0.04	-0.01
TR698	4.65	0.30	0.06	-0.00	0.07	-0.00
TR705	4.71	0.33	-0.07	-0.01	-0.02	-0.00
TR708	4.63	0.10	-0.00	-0.01	0.03	-0.00
TR710	2.44	0.13	-0.06	-0.01	-0.05	-0.01
TR712	1.99	0.06	-0.07	-0.01	-0.08	-0.01
TR722	4.42	0.42	0.06	0.00	0.13	0.01
TR723	2.23	0.12	-0.07	-0.01	-0.16	-0.02
TR724	5.95	0.36	0.09	-0.00	0.05	-0.00
TR738	1.40	0.04	-0.04	-0.00	-0.08	-0.01
TR747	11.95	0.28	0.04	-0.01	0.09	-0.01
TR752	1.50	0.07	-0.05	-0.01	-0.04	-0.01

Table 7.2: Data for the second experiment on the T$_{CASP}$ test set where the convergence limit L is set at 0.5Å. RMSD$_i$ and TM$_i^*$ refers to the initial target RMSD and $1-$TM-score value before minimization. Δ RMSD$_1$ and Δ TM$_1^*$ are the improvements in RMSD and TM-score for the raw potential and Δ RMSD$_2$ and Δ TM$_2^*$ are the improvements in RMSD and TM-score for the decoy-convergence improved potential.

Chapter 8

Designing smooth knowledge-based potentials with local minima in native structures

M. Carlsen, P. Røgen, Designing smooth knowledge-based potentials with local minima in native structures, in submission, 2015.

Abstract. One obstacle to protein structure prediction is that knowledge-based potentials generally do not find the native structure of a protein to be the lowest energy state. We therefore present two methods to design a smooth potential with local minima in a set of native structures. The first method is a tractable relaxation of the usual stability conditions requiring almost vanishing gradient and an almost positive semidefinite Hessian in the desired minima while optimizing a least squares fit to a linear relation between decoy-native energy difference and decoy-native distance on a larger set of protein folds. The second method is based on an iterative strategy where a knowledge based potential first is defined by the same least squares fit. Next, structural optimization of an unstable native structure yields structures with lower energy than the native structure and these are added to the existing decoy library and the next improved potential is constructed. We demonstrate that both methods are capable of sculpting a knowledge-based potential with local minima close to a smaller number of native structures and demonstrate that this is very restrictive for the coarse grained knowledge-based potential. The methods developed here can be used for any smooth potential that is linear in its parameters.

Keywords: Knowledge-based potentials, local minima, optimization, semidefinite programming, funnel sculpting.

8.1 Introduction

The energy landscape of a protein is difficult to predict due to its complexity[127]. While several views on the shape of the energy landscape have emerged, a general agreement dating back to Anfinsen's experiments is that the native protein is a stable global minimum in the energy landscape[128]. Hence, if a potential should be able to be used for protein folding simulations and stability analysis, then the potential must have a minimum when evaluated at, or at least near, a native protein structure. The purpose of this study is to introduce an optimization technique allowing you to enforce local minima in the energy landscape for a set of native protein structures.

It is well-known that the molecular potentials have many local minima due to the roughness of the energy landscape[129] and several searching strategies have been developed to find the true global minimum[130]. Even though these potentials have been used with great success to study protein folding, they are not sufficiently accurate to separate native from non-native structures. Different methods have therefore been developed to design knowledge based potentials which are able to recognize the native (or near native) structure in an ensemble of generated structures. Here, the parameters of the score function are estimated using a linear program[131, 50, 43, 45, 132, 133]. The linear program is primarily a feasibility problem where a score function is constructed from a set of linear inequalities[44]. The size of the energy gab between a native structure and a set of decoys is, however, not optimized. A non-linear z-score minimization problem is solved instead to maximize the energy gab[52, 134, 135]. An excellent review of these methods can be found in ref. [136].

The potentials most often used for optimization are non-smooth discriminators. A Chebyshev expanded and thus smooth potential has been developed for Z-score minimization[121] and for protein folding[57] where they use an iterative method based on the idea that the energy landscape is funnel-shaped[137, 138, 139, 140]. The iterative method cycles between a stochastic search algorithm and a parameter estimation with the end goal of obtaining a potential with maximal funnel smoothness and the property that the search algorithm converges to a native structure. Since the search algorithm is a stochastic Monte-Carlo algorithm, the method they suggest could also have been used for non-smooth potentials where the gradients are not available.

Recently, a quadratic optimization procedure has been suggested that optimizes a knowledge-based potential to have a high correlation to RMSD[12]. While this method was developed with the purpose of training a knowledge-based potential, it can be seen as a method to model the energy gab given that RMSD is sufficiently small. Furthermore, it can be applied to any measure of distance such as The Global Distance Test GDT-TS, the number of native contacts, a spring model such as FlexE[11] or MT[120] or the Scaled-Gauss Metric[141]. The importance of the choice of distance measure in the training of a b-spline expanded pair potential has been investigated in ref. [120].

Here, we introduce a novel method to design a potential that has local minima for a set of native structures. The conditions for a local minimum are a vanishing gradient and a positive semidefinite Hessian. These conditions are also referred to as the first and second order optimality conditions since they are the two convergence criteria for a Hessian-based minimization algorithm. Thus, we require the following conditions to be satisfied:

$$\begin{aligned} \nabla E(\boldsymbol{X})|_{\text{native}} &= 0 \\ \nabla^2 E(\boldsymbol{X})|_{\text{native}} &\succeq 0, \end{aligned} \tag{8.1}$$

where E is a potential, $\boldsymbol{X}$ is vector containing our 2000 model parameters and the symbol $\succeq$ indicates that the matrix is positive semidefinite. This leads to a semidefinite problem due to equation (8.1) (the Hessian cannot be positive definite as the energy of a protein is independent of translation and rotation). We use a relaxation of the original problem where the norm of gradient and the size of the negative eigenvalues of the energy potential are kept small. This relaxation allow us to decouple the coarse and fine grained parts of the potential and reduces the size of the optimization problem significantly. Further away from the native structures the potential is sculpted using same method as in ref. [12, 120]. Finally, we introduce a fast iterative method to approximately stabilize a set of native structures. Starting with a potential sculpted as in ref. [12, 120] we perform energy minimization of the native structure. If the native structure is not stable this leads to a structure with lover energy. We then add this structure to the pool of decoy-structures and re-sculpt the potential. In the re-sculpting it is favorable to raise the potential in the new decoy and the worst escape direction from the native structure in the previous potential is now changed.

8.2 Methods

8.2.1 The local and global potential

We use the knowledge-based potential described in Ref. [142]. The knowledge-based potential consists of three parts: a backbone potential, a hydrogen potential and a global potential. The backbone model consists of harmonic terms for the main atoms in the backbone (N, H, C_α, C and O). We thus only allow small variations of the bond lengths, bond angles and the dihedral angles about the peptide bonds which means that our model restricts motions to be close to those of a dihedral angle model. The hydrogen potential is harmonic potential defined on the $O-H$ bonds, the bond angles $N-H-O$ and $H-O-C$ and the torsion angle $N-H-O-C$ for each hydrogen bond. We refer to the sum of the backbone potential and the hydrogen potential as the local potential E_{L}. Our coarse grained non-bonded global potential, E_{G}, consists of a pair potential and a solvent potential and is defined on the C_α atoms and expanded in terms of b-spline basis functions.

8.2.2 Formulation of the optimization problem

Our aim is to estimate the energy gab for the global potential $E_G(\boldsymbol{X})$ with the parameters $\boldsymbol{X}$ given that the criteria in equation (8.1) are satisfied. We use the metrics NT, $Q9$ and $Q5_{i,j}$ described in Ref. [142]. Let the subindex i refer to the i-th native structure and indices i,j refer to the j-th decoy of the i-th native structure. The objective function is a quadratic function given by

$$f(\boldsymbol{X}) = \sum_{i,j} \|E_{G_{i,j}}(\boldsymbol{X}) - E_{G_i}(\boldsymbol{X}) - NT_{i,j} - \alpha_i \cdot Q9_{i,j} - \beta_i \cdot Q5_{i,j}\|^2 + \gamma \cdot \|\boldsymbol{X}\|^2, \qquad (8.2)$$

where the last term is added since we are looking for a Tikhonov regularized solution which ensures that the quadratic matrix has full rank. We fix γ at 10^{-2} and fix the constants α_i and β_i by a linear fit, $NT_{i,j} \approx \alpha_i \cdot Q9_{i,j}$ and $NT_{i,j} \approx \beta_i \cdot Q9_{i,j}$ for each i. This means that for each decoy ensemble all of the metrics have the same energy scale.

Furthermore, we require that the first parameter for the pair potential in each b-spline expansion for the pair potential is non-negative. The first criterion is not difficult to introduce as it leads to an ordinary least square with linear constraints. We require that the Hessian in the i-th native structure is positive semidefinite to ensure that our potential satisfies the second criterion:

$$\nabla^2 E_i = \nabla^2 E_{L_i} + \nabla^2 E_{G_i}(\boldsymbol{X}) \succeq 0, \tag{8.3}$$

where $\nabla^2 E_{G_i}$ and $\nabla^2 E_{L_i}$ are the Hessian of the global and the local potential for the i-th native structure. The non-linear semidefinite optimization problem is thus

$$\begin{aligned} \underset{\boldsymbol{X}}{\text{minimize}} \quad & f(\boldsymbol{X}) \\ \text{subject to} \quad & \nabla E_{G_i}(\boldsymbol{X}) = 0,\ i = 1 \dots N \\ & \nabla^2 E_{L_i} + \nabla^2 E_{G_i}(\boldsymbol{X}) \succeq 0,\ i = 1 \dots N, \end{aligned} \tag{8.4}$$

where N is the number of native structures. The formulas for the gradient and Hessian of an internal coordinate, and thus of E_{L_i}, can be found in Ref. [125]. It is preferable to use a relaxation of the problem where the gradient and the negative eigenvalues are kept small. This leads to the relaxed SDP :

$$\begin{aligned} \underset{\boldsymbol{X}}{\text{minimize}} \quad & f(\boldsymbol{X}) \\ \text{subject to} \quad & \|\nabla E_{G_i}(\boldsymbol{X})\| \leq \epsilon,\ i = 1 \dots N \\ & \nabla^2 E_{L_i} + \nabla^2 E_{G_i}(\boldsymbol{X}) \succeq -\epsilon I,\ i = 1 \dots N \end{aligned} \tag{8.5}$$

where ϵ is set to 10^{-5}. We use the optimization package SDPARA [143] which solves a linear semidefinite problem (see below) utilizing parallel computing. The gradient and Hessian of the local and global potential have been calculated using the algorithms and formulas in [125]. We introduce two reformulations of the problem to a form that has better convergence properties and where we can use parallel computing.

8.2.3 Reformulation - 1

The program SDPARA which uses parallel computing requires the objective function to be linear. Hence, we have to reformulate the optimization problem to an equivalent semidefinite problem with a linear objective function. Usually, it is not preferable to reformulate a quadratic problem to a semidefinite problem as it takes longer time to solve the equivalent semidefinite problem (if it can be solved) even though the solution is the same[144]. Here, however, we solve the optimization problem for a large number of proteins and thus a large number of linear matrix inequalities. We expected that to solve such a large problem it would pay off to reformulate the problem to an equivalent semidefinite problem where parallel computing is used.

In the following, we sketch how a quadratic problem can be reformulated to a semidefinite problem. We consider the unconstrained quadratic problem

$$\underset{\boldsymbol{X}}{\text{minimize}} \quad f(\boldsymbol{X}), \tag{8.6}$$

where $\boldsymbol{X}$ is a variable and the objective function $f(\boldsymbol{X})$ is a quadratic form

$$f(\boldsymbol{X}) = \boldsymbol{X}^T B \boldsymbol{X} + \boldsymbol{X}^T \boldsymbol{c} + d. \tag{8.7}$$

The matrix B has a Cholesky decomposition, $B = A^T A$, since B is positive definite and f can thus be written as

$$f(\boldsymbol{X}) = (A\boldsymbol{X} - \boldsymbol{b})^T (A\boldsymbol{X} - \boldsymbol{b}) + constant, \tag{8.8}$$

such that $\boldsymbol{c} = -2A^T\boldsymbol{b}$. The last term is not important and the original problem is thus equivalent to an ordinary least square problem

$$\underset{\boldsymbol{X}}{\text{minimize}} \quad \|A\boldsymbol{X} - \boldsymbol{b}\|_2^2. \tag{8.9}$$

This problem is equivalent to:

$$\begin{aligned} &\underset{\boldsymbol{X},t}{\text{minimize}} \quad t \\ &\text{subject to} \quad \|A\boldsymbol{X} - \boldsymbol{b}\|_2^2 \leq t. \end{aligned} \tag{8.10}$$

It can be shown that (see for instance Ref. [145])

$$t - \|A\boldsymbol{X} - \boldsymbol{b}\|_2^2 \geq 0 \text{ iff. } \begin{pmatrix} I & A\boldsymbol{X} - \boldsymbol{b} \\ (A\boldsymbol{X} - \boldsymbol{b})^T & t \end{pmatrix} \succeq 0. \tag{8.11}$$

The problem above is therefore equivalent to the semidefinite problem[144].

$$\begin{aligned} &\underset{\boldsymbol{X},t}{\text{minimize}} \quad t \\ &\text{subject to} \quad \begin{pmatrix} I & A\boldsymbol{X} - \boldsymbol{b} \\ (A\boldsymbol{X} - \boldsymbol{b})^T & t \end{pmatrix} \succeq 0, \end{aligned} \tag{8.12}$$

with the variables $\boldsymbol{X}$ and the new variable t.

8.2.4 Reformulation - 2

The linear matrix inequalities in equation (8.5) can be reformulated to an equivalent form which reduces the number of elements in each matrix by a factor of 5×5. The price we pay is that the matrices become more dense. Here, we will only consider amino acids with 5 backbone atoms which means that we exclude proline which only has 4 backbone atoms. Each of our linear matrix inequalities have the partitioned matrix form

$$\nabla^2 E_{L_i} + \nabla^2 E_{G_i}(\boldsymbol{X}) + \epsilon I = \begin{pmatrix} H_{L_{i,1}} + \epsilon I & H_{L_{i,2}} \\ H_{L_{i,2}}^T & H_{L_{i,3}} + H_{G_i}(\boldsymbol{X}) + \epsilon I \end{pmatrix} \succeq 0, \tag{8.13}$$

where $H_{L_{i,1}}$ depends on the N, H, C and O atoms, $H_{L_{i,2}}$ on the N, H, C_α, C and O atoms, and $H_{L_{i,3}}$ and H_{G_i} on the C_α atoms in the backbone of a protein. Since $H_{L_{i,1}}$ is positive definite, it can be shown that

$$\nabla^2 E_{L_i} + \nabla^2 E_{G_i}(\boldsymbol{X}) + \epsilon I \succeq 0 \text{ iff. } H_{G_i}(\boldsymbol{X}) + H_{L_{i,3}} + \epsilon I - H_{L_{i,2}}^T \left(H_{L_{i,1}} + \epsilon I\right)^{-1} H_{L_{i,2}} \succeq 0. \tag{8.14}$$

The matrix $H_{G_i}(\boldsymbol{X}) + H_{L_{i,3}} + \epsilon I - H_{L_{i,2}}^T \left(H_{L_{i,1}} + \epsilon I\right)^{-1} H_{L_{i,2}}$ is referred to as Schur's complement. The optimization problem can thus be formulated as

$$\begin{aligned} &\underset{\boldsymbol{X}}{\text{minimize}} \quad f(\boldsymbol{X}) \\ &\text{subject to} \quad \|\nabla E_{G_i}(\boldsymbol{X})\|_\infty \leq \epsilon, \ i = 1 \ldots N \\ &\qquad\qquad H_{G_i}(\boldsymbol{X}) + H_{L_{i,3}} + \epsilon I - H_{L_{i,2}}^T \left(H_{L_{i,1}} + \epsilon I\right)^{-1} H_{L_{i,2}} \ \succeq 0, \ i = 1 \ldots N. \end{aligned} \tag{8.15}$$

The size of the matrices in the linear matrix inequalities are reduced by a factor of 5×5 since each of the $H_{G_i}(\boldsymbol{X})$ matrices is a $3n \times 3n$ matrix involving only C_α atoms where n is number of amino acids in the protein. Using the first reformulation above, the final relaxed SDP form is:

$$\begin{aligned} \underset{\boldsymbol{X},t}{\text{minimize}} \quad & t \\ \text{subject to} \quad & \begin{pmatrix} I & A\boldsymbol{X} - \boldsymbol{b} \\ (A\boldsymbol{X} - \boldsymbol{b})^T & t \end{pmatrix} \succeq 0. \\ & \|\nabla E_{G_i}(\boldsymbol{X})\|_\infty \leq \epsilon, \ i = 1 \ldots N \\ & H_{G_i}(\boldsymbol{X}) + H_{L_{i,3}} + \epsilon I - H_{L_{i,2}}^T \left(H_{L_{i,1}} + \epsilon I\right)^{-1} H_{L_{i,2}} \ \succeq 0, \ i = 1 \ldots N, \end{aligned} \tag{8.16}$$

with the variables $\boldsymbol{X}$ and t.

8.2.5 An iterative method to generate a better data set

The training set we use, see Data sets, consists mainly of decoys with an RMSD value between 2 and 3 Angstrom. Since the training set lacks decoys within a distance of 2 Angstrom, it is not surprising that it does not lead to a potential which when evaluated at a native structure has positive semidefinite Hessian and a vanishing gradient. We, therefore, asked the question whether it is possible to obtain a potential with the desired local properties if we use a fined-tuned training set instead.

With the purpose of investigating this, we introduce an iterative method to generate a new training set which consists of near-native decoys with a RMSD less than 1 Angstrom. The new training set is generated in the following way: First we determine the parameters to the global potential with a least square optimization using the original training set. Next, we start a minimization algorithm from a native structure until it is at a RMSD distance of about $1\mathring{A}$ or the difference in function value is less than 10^{-6}. From this trace of decoys we pick a set of 20 decoys. We are interested in having a distribution of decoys with RMSD distance between 0 to 1 Angstrom so the decoys have been chosen as uniformly as possible. The decoys are added to the training set and we thereby obtain a new training set with decoys having RMSD values less than $1\mathring{A}$. The new decoys are then added to the decoys with a weight factor of 100. When reoptimizing the potential its favorable to raise the energy of the new decoys if possible and hereby hopefully close the worst possible direction of the previous potential. The algorithm is continued iteratively. The minimization algorithm used to energy minimize protein structures in this study is a Hessian-based algorithm that uses directions of negative curvature [58, 59].

Finally, we remark, that the spring constants for the hydrogen potential are so low that it for the iterative method is possible to stretch a hydrogen bond during a minimization such that the bond angles take values close to π. There may therefore be a risk of dividing by zero. We decided to turn off the bond angle and torsion angle potentials for the hydrogen potential to ensure numerical stability of the minimization routine which thereby is a O-H distance potential only.

8.2.6 Data sets

We use a training set taken from the Titan-High Resolution Decoys which for 1400 non-homologous proteins are generated by the torsion angle dynamics program DYANA41

Table 8.1: The orthogonal bundle alpha-helix proteins used in this study.

Name	CATH3.5	Size
1eijA	1-10-8-140	72
1awcA	1-10-10-10	110
1ahdP	1-10-10-60	68
1aa7A*	1-10-10-180	158
1ji8A*	1-10-10-370	111
1dp3A	1-10-10-450	55
1a7wA	1-10-20-10	68
1hryA	1-10-30-10	73
1afhA	1-10-110-10	93
1b1uA	1-10-120-10	117
1cmaA	1-10-140-10	104
1cooA	1-10-150-20	81
1b0xA	1-10-150-50	72
1d8bA	1-10-150-80	81
1ecwA	1-10-150-90	114
1bnoA	1-10-150-110	87
1fiqA*	1-10-150-120	164
1agrE*	1-10-167-10	128
1ccdA	1-10-210-10	77
1bo9A	1-10-220-10	73

where the hydrophobic core in a protein are subject to constraints[47, 107]. Here we use a non-homologous subset of 20 decoy ensembles that either belong to the orthogonal bundle alpha-helix class or in the case of the *-marked chains the orthogonal bundle constitute only one subdomain of the chain. Their pdb-code can be found in Table 8.1.

8.3 Results

An important property of a protein potential is that the native structure is a global minimum. We propose two methods to generate a potential with local minima for a set of native structures: 1) A method based on semidefinite programming where the local minima explicitly is included in the formulation of the optimization problem, where we use a tractable relaxation of the original problem. We, hereby, permit that the potential when evaluated at a native structure has a saddle point with small negative eigenvalues. The purpose of this method is thus to keep the norm of the gradient and the negative eigenvalues of the Hessian of the potential small for a set of native structures. This leads to the relaxed semidefinite optimization problem given by equation (8.16). 2) An iterative method where optimizing the energy surface as measured in a large set of models of a set of proteins is alternated with generation of additional models through structural optimization of the native models in the constructed energy surface.

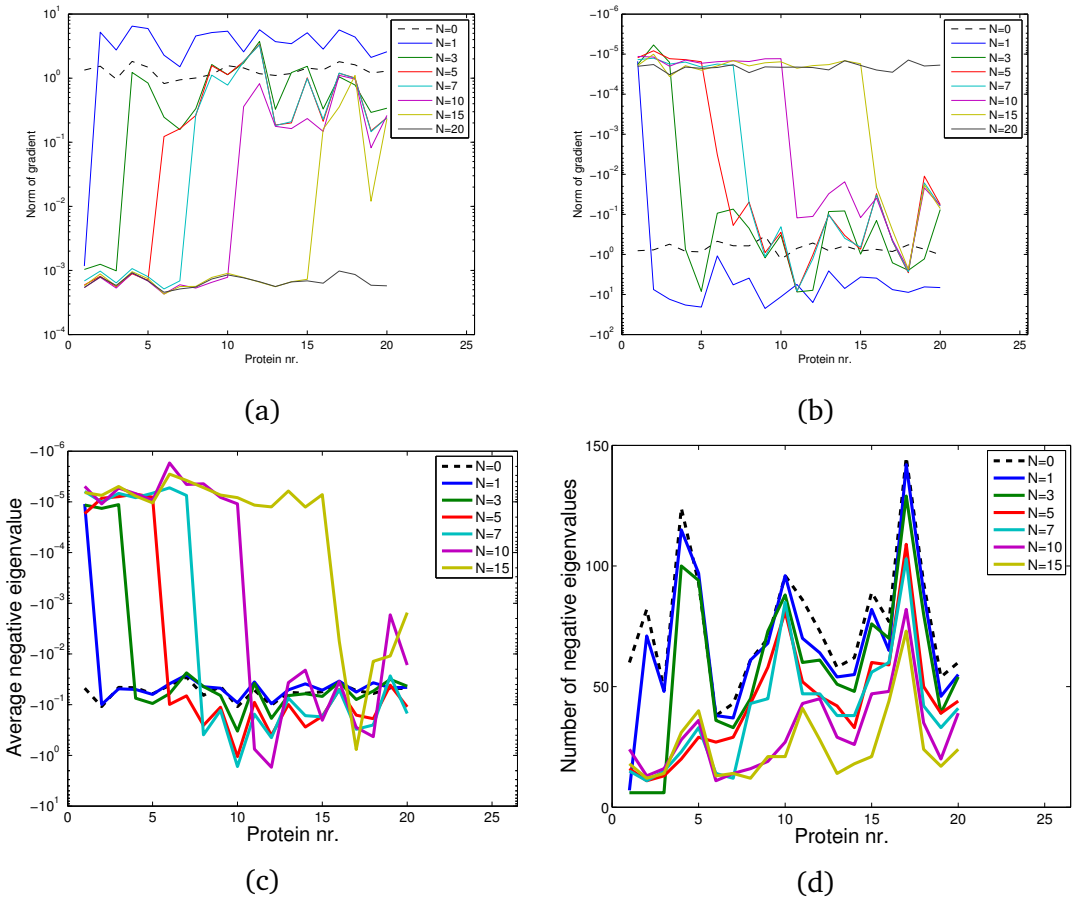


Figure 8.1: We have solved the optimization problem in equation (8.16) with $N = 0, 1, 3, 5, 7, 10$ and 15 conditions. For each of these, we have plotted the norm of the gradient (a), the lowest eigenvalue (b), the average negative eigenvalue (c) and the number of negative eigenvalues (d). There is a decrease by four orders of magnitude for the proteins that locally have been optimized with the SDP-method.

8.3.1 Using the semidefinite programming method

We consider a set of 20 alpha-helix proteins to investigate how effective the method is. We solve the optimization problem while enforcing the relaxed minimality conditions (equation 8.16) for the first $N = 0, 1, 3, 5, 7, 10$ and 15 proteins in the set. For each training the negative eigenvalues of the Hessian and the norm of the gradient for all 20 proteins are illustrated in Fig. 8.1a and 8.1b. It is seen that the size of the gradient and the lowest negative eigenvalue of the energy potential are roughly 10^{-4} and -10^{-5} for the optimized native structures corresponding to a decrease with the factor of 10^4. The number and average of the negative eigenvalues are shown in Fig. 8.1c and 8.1d. They both have a notable decrease, as expected. The method thus leads to a potential whose Hessian when evaluated in one of the N native structures with active constraints has small and few negative eigenvalues. We used Pearson's correlation coefficient to compare the optimal parameter sets found for $N = 0, 1, 3, 5, 7, 10$ and 15. These can be found in Table 8.2. The correlation to the $N = 0$ parameter set decreases significantly as more proteins are added to the problem. The constrains on the solution space which increase with N thus lead to significant changes in the solutions to the

Table 8.2: Correlation between parameters.

M	0	1	3	5	7	10	15
0	1	0.77	0.52	0.33	0.30	0.29	0.28
1	0.77	1	0.59	0.32	0.27	0.30	0.30
3	0.52	0.59	1	0.63	0.56	0.52	0.55
5	0.33	0.32	0.63	1	0.97	0.82	0.59
7	0.30	0.27	0.56	0.97	1	0.82	0.55
10	0.29	0.30	0.52	0.82	0.82	1	0.72
15	0.28	0.30	0.55	0.59	0.55	0.72	1

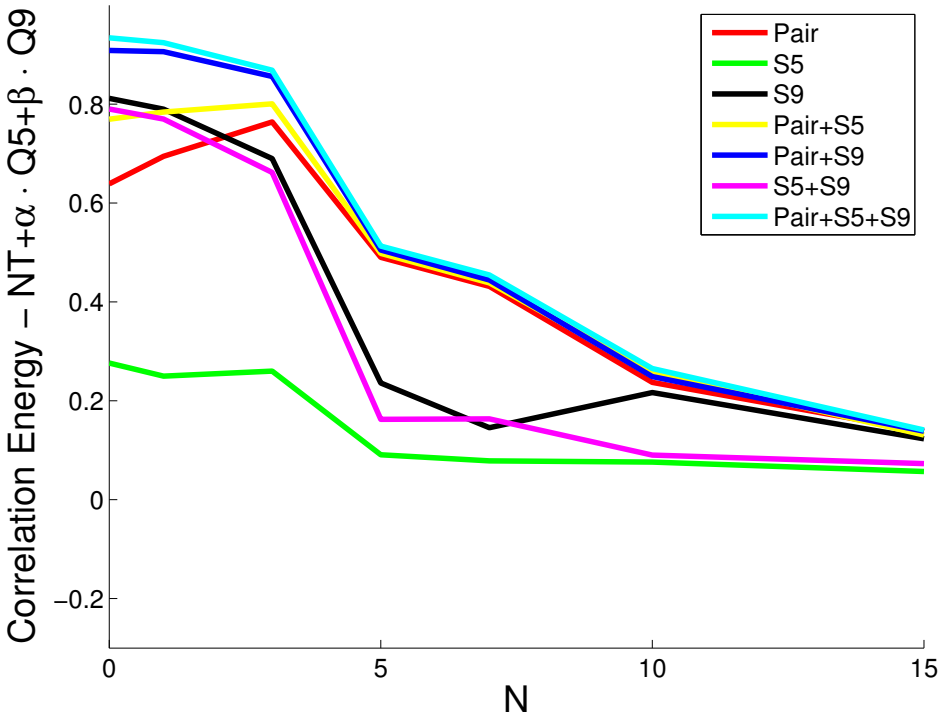
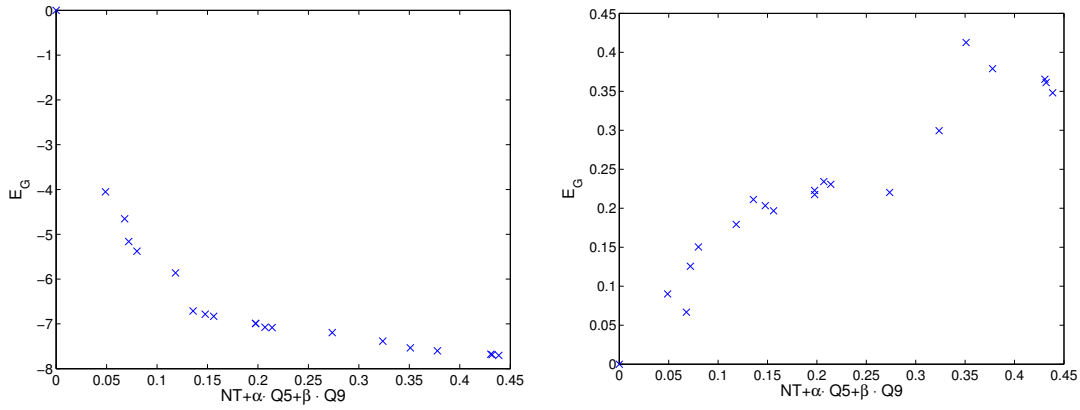


Figure 8.2: Showing the correlation between different components of the energy and a metric. All values are averaged over 20 different protein ensembles. The full potential has the highest correlation to all of the metrics except C5. Clearly, there is a drop in correlation when we add more constrains to the problem.

optimization problem. To investigate this further, we analyzed the average correlation of the 20 protein ensembles used in the training of the potentials between four metrics and the produced individual components of the potential as shown in Fig. 8.2. For the full potential we find that the correlation decreases as N increases, as we would expect since the solution space is reduced when we add more constrains to the problem. The pair potential and the solvent potential E_{S9} have the highest correlation to the metric. Removing any of the two potentials leads to a significantly lower performance for small N. For $N = 7$ the pair potential dominates. For N greater than 10 the average correlation is about the same for the two potentials. Apparently, the solvent potential E_{S5} does not contribute as much as the pair potential and the solvent potential E_{S5}



(a) (b)

Figure 8.3: (a) The global energy of a set of decoys obtained by a structural optimization from a native structure as function of their distance from the native structure. (b) After re-optimizing the potential energy function the energy of these new decoys is raised and correlate and scale well with the decoy-native distance.

to the performance for any N. The difference is small when comparing different the metrics $NT_{i,j} + \alpha_i \cdot Q5_{i,j} + \beta_i \cdot Q9_{i,j}$, $NT_{i,j}$ and $Q9_{i,j}$ (data not shown).

8.3.2 Using the iterative method

The principle behind the iterative method is to generate a new and improved training set where the energy of a set of near-native decoys which have a lower energy than the energy of the native structure is raised. This is shown in Fig. 8.3a and 8.3b. The algorithm shifts between a searching algorithm and a parameter optimization. At each step we select a set of 20 near-native decoys within a RMSD distance of $1\mathring{A}$ from a structural optimization started at a native structure. These decoys are then added to the existing training set and we require in the following parameter optimization that their energy is in concordance with the targets i.e. NT, Q_5 and Q_9. The decoys have a lower energy than the native structure since we picked them from a minimization routine, see Fig. 8.3a. As a consequence, the energy is adjusted in the following energy optimization, see Fig. 8.3b. The algorithm continues until the change in the parameter values is sufficiently small. Fig. 8.4 shows the L2-norm between each parameter set for 200 iterations for a single protein. We see that the change in the parameter values after roughly 50 steps are small and insignificant. We note that this number decreases with the weight factor that we use in the parameter optimization and increase with the number of proteins considered in the iterative method. We have calculated the norm of the energy gradient as well as the lowest eigenvalues, the average eigenvalues and the number of negative eigenvalues of the Hessian for a single protein as shown in Fig. 8.5. We see that there is a small change in the values after roughly 50 steps in concordance with the fact that the parameters have a small variation after about 50 steps. Furthermore, the gradient and the lowest negative eigenvalue decreases by roughly 2 orders of magnitude. The iterative method may just as well be used for more than one protein such that a set of 20 near-native decoys for each of the $N = 0, 1, 3, 5, 7, 10$ or 15 proteins are added to the trainings set. We have calculated the

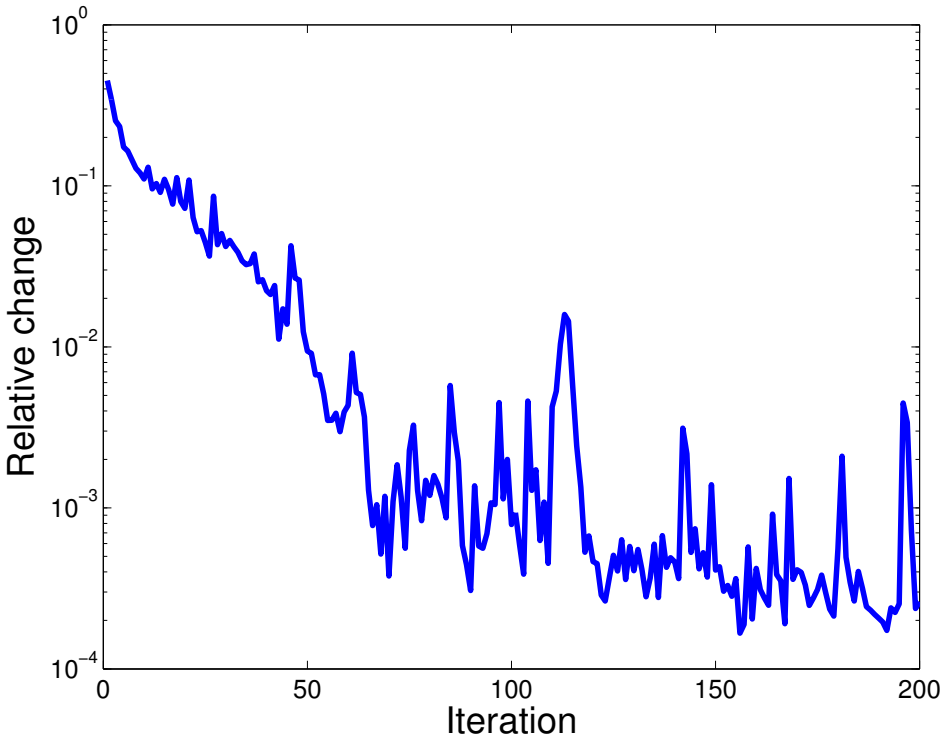


Figure 8.4: For each iteration, we have plotted the relative change $\frac{\|\boldsymbol{X}_i - \boldsymbol{X}_{i+1}\|_2}{\|\boldsymbol{X}_i\|_2}$ between the i-th parameter set $\boldsymbol{X}_i$ and the $i+1$-th parameter set $\boldsymbol{X}_{i+1}$.

norm of the gradient as well as the lowest eigenvalues, the average eigenvalues and the number of negative eigenvalues of the Hessian for each of these after 200 iterations in Fig. 8.6. Clearly, 200 iterations is more than enough for one protein. However, we decided to use 200 iterations as it was sufficient for convergence for all N considered here. In agreement with the results for a single protein shown in Fig. 8.4, the norm of the gradient and the magnitude of the lowest negative eigenvalue drops two orders of magnitude. In Fig. 8.7 we have plotted the average correlation between the metric and the individual components of the potential. The figures are almost identical to when we used the semidefinite programming method shown in Fig. 8.2 and the conclusions are therefore the same as for the semidefinite method.

8.4 Discussion

We have presented two methods to design smooth knowledge-based protein potentials with simultaneous local minima in each of a smaller set of native protein structures while still shaping the potential in a larger number of different protein folds. At present the main reason for doing this is to provide a tool that can investigate if a given functional form of protein potentials can stabilize a set of native structures and quantify how restrictive it is for a family of potentials.

We model our protein potential in Cartesian coordinates but use a bonded potential for hydrogen bonds and for the main atoms in the backbone where it allows only small variations of the bond lengths, bond angles and the dihedral angles about the peptide bonds and thus restrict motions to be close to those of a dihedral angle model. Our

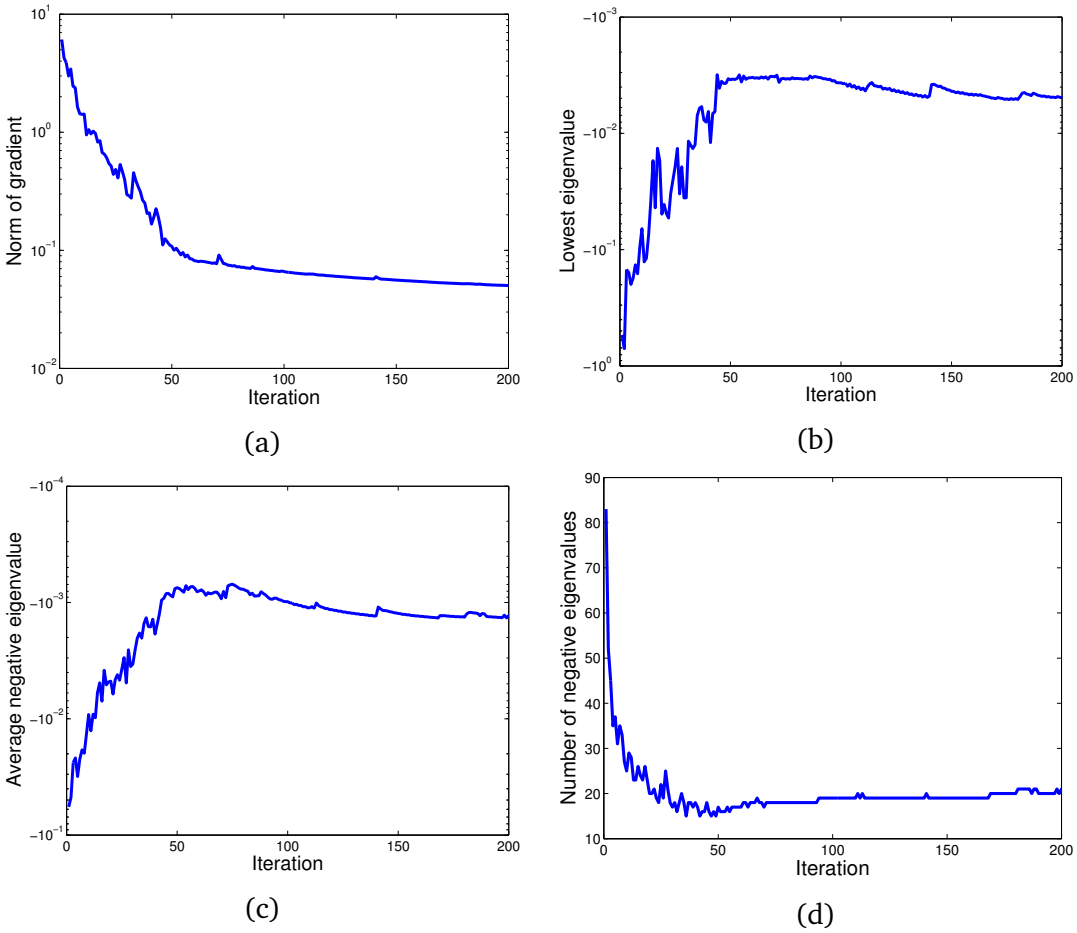


Figure 8.5: For the iterative method the norm of the gradient (a), the lowest negative eigenvalue (b), the average negative eigenvalue (c) and the number of negative eigenvalues (d)are shown as function of the number of iterations. The decrease in the values is most significant for the first 50 iterations.

coarse grained model is similar to the most used carbon alpha distance based pair and solvent potentials, but as we need explicit Hessian's, our model is two times continuous differentiable and is b-spline expanded with 2000 parameters.

Whereas the functional form of our potential is kept similar to basic knowledge based protein potentials the choice of its parameters are not. All parameters of the pair and solvent potential are given by so-called metric training where basically the correlation between decoy-native energy differences and decoy-native distances is optimized simultaneously for a larger set of native structures and decoys of these. Hence, restricted to a given set of native and decoy structures metric training picks the best performing potential with the given functional form.

The direct approach to enforce a local minimum in a given native structure is to require vanishing gradient and positive semidefinite Hessian. We present a relaxation of this demand that allow us to decouple the coarse grained pair and solvation potential from the local bonded potential reducing the size of the optimization problem with a factor of 5×5. In the relaxed formulation we enforce a gradient with small norm and a Hessian with numerically small negative eigenvalues in each native structure which perhaps may be justified by the final resolution of the native structure. We use

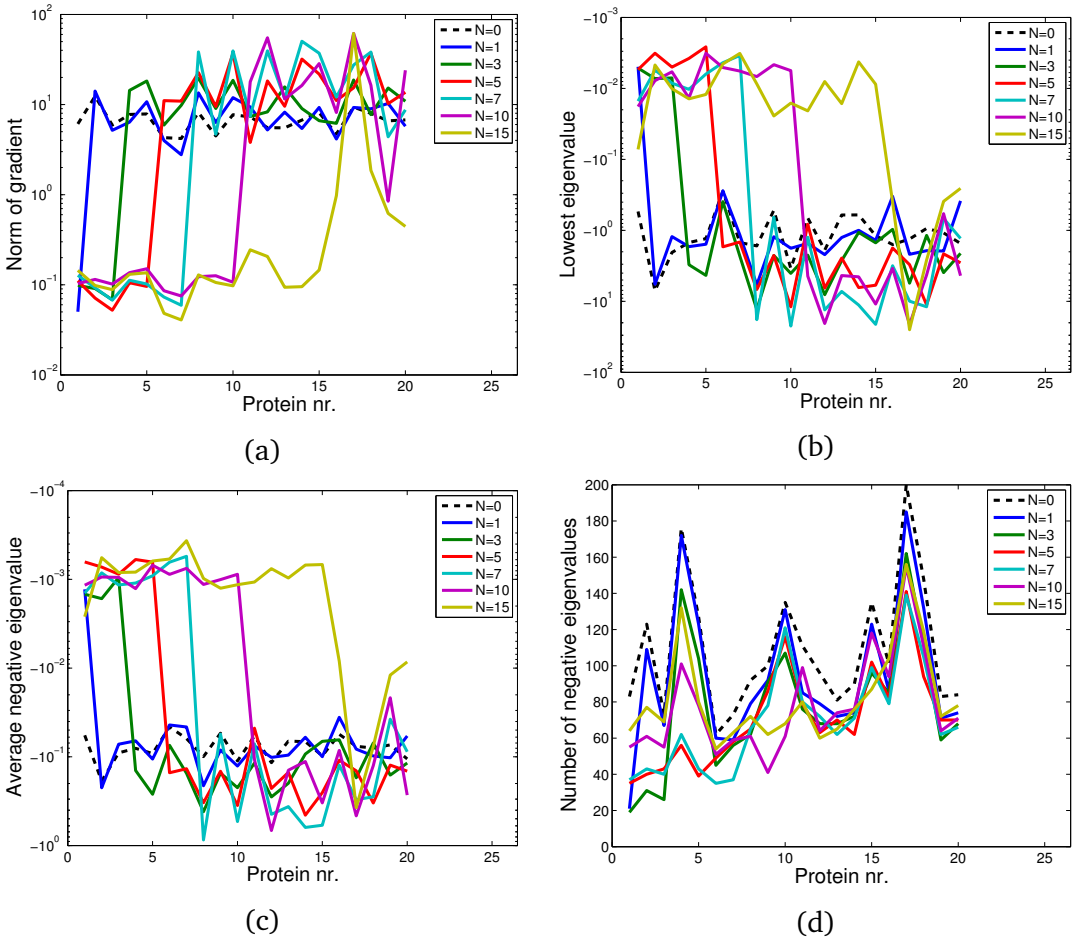


Figure 8.6: We have plotted four figures for $N = 0, 1, 3, 5, 7, 10$ and 15 proteins after 200 iterations: the norm of the gradient (a), the lowest eigenvalue (b), the average negative eigenvalue (c) and the number of negative eigenvalues (d). We observe a decrease of two orders of magnitude for the proteins that have been optimized with the iterative method.

this relaxed formulation as constraints to the metric training of the potential which unconstraint typically gives linear correlations between decoy-native energy differences and decoy-native distances of 0.89 for near native decoys.

The method works in the sense that we manage to decrease the norm of the gradient and the absolute value of the negative eigenvalue of the Hessian with a factor of 10^4. While we are pleased with this performance, it may not be optimal performance since we had to lower the accuracy of the optimal solution to ensure convergence of the solver. A better performance may be achieved when the SDP solvers become more stable and accurate.

The second method starts with a potential metrically trained on a set of native and decoy structures. Each native structure is energy minimized using this potential and the resulting structures are added to the set of decoys and the potential is re-trained. When repeating this procedure the potential parameters seems to converge and norm of the native gradient and the absolute value of the negative eigenvalue of the native Hessian decrease with at factor of 10^2 which is quite efficient when taking in to consideration that the method has no direct control of gradients and Hessians. Hence, the iterative

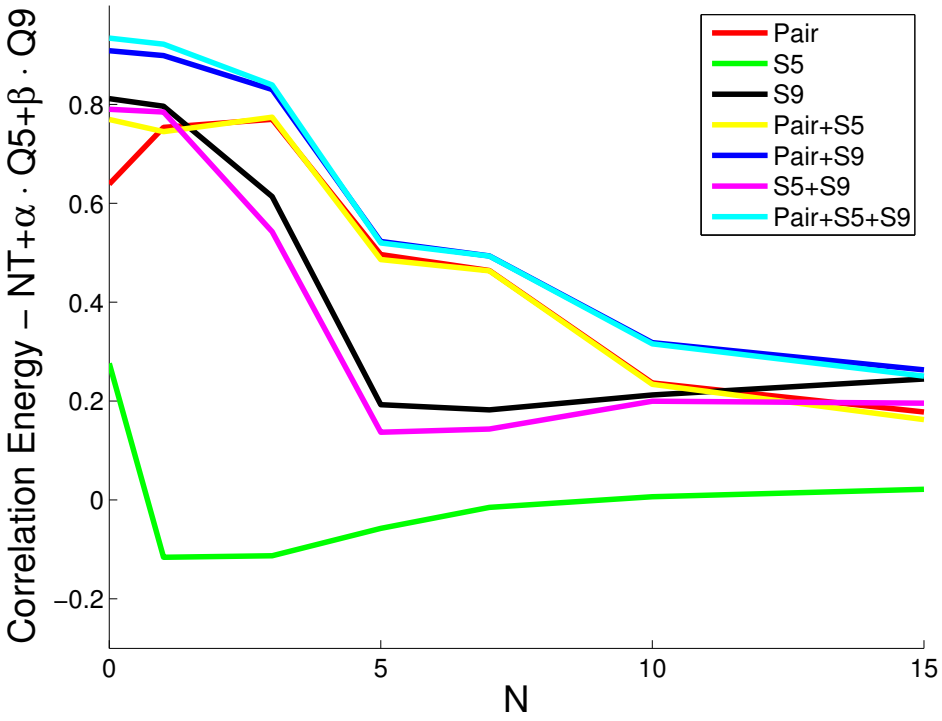


Figure 8.7: Showing the correlation between different components of the energy and a metric where we have used the iterative method. The results are almost identical to those obtained using the semidefinite method as shown in Fig. 8.2.

strategy works relatively efficient and has the benefit of being computationally simpler than the semidefinite method.

Having established two methods to approximately stabilize a set of native structures we investigate the important question, whether it is possible to sustain the high correlation between decoy-native energy difference and distance while we require the constrains to be satisfied. Both our methodologies suggest that this is possible for only for a few native proteins since the average energy-distance correlation decreases from 0.93 to 0.92 to 0.87 to 0.51 when 0 (zero), 1, 3 and 5 native structures are stabilized respectively. This indicates that it is very restrictive to stabilize non homologous native structures for a potential with the given functional form, and points in the direction of redoing the experiment for a more detailed model e.g. with a coarse grained potential that is not spherical symmetric. Another interesting application is to investigate how many sequence homological structures that can be stabilized with the same potential.

8.5 Acknowledgements

The authors want to thank Mathias Stolpe for discussions and careful reading of the manuscript.

Conclusions and future work

In this thesis, we have investigated the problem of assessing and refining the quality of predicted protein structures using knowledge-based potentials. The main purpose of this thesis is to develop techniques to improve the decoy convergence of our knowledge-based potential. We succeeded in doing this for shorter energy minimizations in a metrically trained potential using an iterative strategy that resulted in a potential that is less dependent on the initial training set. We found an improvement in performance by using a metric based on intrinsic geometry and analyzed two methods for finding the optimal metrically trained potential that simultaneous has a number of native structures as a local minimum. We also derived new formulas to calculate the first- and second-order derivatives of a molecular potential that can be implemented with high efficiency in high-level programming languages based on vectorization. All in all, we have developed several methodologies that significantly improve the performance of our knowledge-based potential. While we are pleased with the performance of our current knowledge-based potential, our results suggest that it has to be modified for it to be more competitive with state-of-the-art quality assessment and quality refinement methods.

The performance may be improved by expanding the functional form of our current knowledge-based potential. The potential is dependent on five atoms for each amino acid in the backbone of the protein. We have thus ignored the atoms in the side chains. The spherical symmetry of our current potential may have to be broken using a side-chain dependent potential that is defined on the center of mass of the side-chain and resembles a half-sphere model. Other possible extensions are a local L-DE potential or a coupling potential. The two side-chain potentials, the local L-DE potential and the coupling potential have been presented in this thesis.

Throughout this work, our knowledge-based potential has been trained on the Titan-HRD training set. It consists of non-homologous proteins and is generated using torsion angle dynamics by adding constrains to the distance between the hydrophobic and hydrophilic amino acids. Using the same generating procedure we may form a training set of homologous proteins thus focusing on improving the quality of structures predicted by comparative modelling. Although this narrows the range of applications to homological models it most likely will improve the performance.

The applications of the methodologies developed during this study reach beyond the uses demonstrated here. Our potential is expanded in terms of cubic b-spline basis functions but the methods apply to any linear expansion of a knowledge-based potential such as a Chebyshev expansion. In principle also a nonlinear expansion although this complicates the optimization procedure as it then cannot be written as a least square. We used a modified Newton method as our potential energy minimization procedure that allowed us to fold near-native proteins relatively fast and to improve the decoy-convergence and stability of our knowledge-based potential. This method which relies

on the first and second derivatives of the potential may be replaced with any energy minimization method such as the L-BFGS method or a stochastic method.

Finally, it is our hope that the different parts of our knowledge-based potential may serve as components in a molecular potential. That would make them more accessible to the protein structure prediction community.

References

[1] C. Anfinsen. Principles that govern the folding of protein chains. *Science*, 181:223–230, 1973.

[2] Y. Zhang. Progress and challenges in protein structure prediction. *Curr. Opin. Struct. Biol.*, 18(3):342–348, 2008.

[3] D. Baker and A. Sali. Protein structure prediction and structural genomics. *Science*, 294(5540):93–96, 2001.

[4] Y. Zhang. Protein structure prediction: when is it useful? *Curr. Opin. Struct. Biol.*, 19:145–155, 2009.

[5] A. Kryshtafovych, A. Barbato, K. Fidelis, B. Monastyrskyy, T. Schwede, and A. Tramontano. Assessment of the assessment: Evaluation of the model quality estimates in casp10. *Proteins: Struct. Funct. Bioinf.*, 82(S2):112–126, 2014.

[6] T. Nugent, D. Cozzetto, and D. T. Jones. Evaluation of predictions in the casp10 model refinement category. *Proteins: Struct. Funct. Bioinf.*, 82(S2):98–111, 2014.

[7] A. McLachlan. Gene duplications in the structural evolution of chymotrypsin. *J. Mol. Biol.*, 128:49–80, 1979.

[8] B. Horn. Closed form solution of absolute orientation using unit quaternions. *J. Opt. Soc. Am.*, 4:629–642, 1987.

[9] E. Coutsias, C. Seok, and K. Dill. Using quaternions to calculate RMSD. *J. Comput. Chem.*, 25:1849–1857, 2004.

[10] A. Zemla. LGA: a method for finding 3D similarities in protein structures. *Nucl. Acids. Res.*, 31:3370–3374, 2003.

[11] A. Perez, Z. Yang, I. Bahar, K. Dill, and J. MacCallum. FlexE: using elastic network models to compare models of protein structure. *J. Chem. Theory Computat.*, 8:3985–3991, 2012.

[12] P. Røgen and P. Koehl. Extracting knowledge from protein structure geometry. *Proteins: Struct. Funct. Bioinf.*, 81:841–851, 2013.

[13] A. Kolinski and J. Skolnick. Reduced models of proteins and their applications. *Polymer*, 45:511–524, 2004.

[14] A. Solis and S. Rackovsky. Optimized representations and maximal information in proteins. *Proteins: Struct. Funct. Bioinf.*, 38:149–164, 2000.

[15] E.-H. Yap, N. L. Fawzi, and T. Head-Gordon. A coarse-grained α-carbon protein model with anisotropic hydrogen-bonding. *Proteins: Struct. Funct. Bioinf.*, 70(3):626–638, 2008.

[16] H. B. Thompson. Calculation of cartesian coordinates and their derivatives from internal molecular coordinates. *J. Chem. Phys.*, 47(9):3407, 1967.

[17] J. Pesonen and K. O. Henriksson. Polymer conformations in internal (polyspherical) coordinates. *J. Comput. Chem.*, 31(9):1873–1881, 2010.

[18] S. Lee and G. S. Chirikjian. Pose analysis of alpha-carbons in proteins. *Int. J. Robot. Res.*, 24(2-3):183–210, 2005.

[19] K. Dill. Polymer principles and protein folding. *Protein Science*, 8:1166–1180, 1999.

[20] M. A. Martí-Renom, A. C. Stuart, A. Fiser, R. Sánchez, F. Melo, and A. Šali. Comparative protein structure modeling of genes and genomes. *Annu. Rev. Biophys. Biomol. Struct.*, 29(1):291–325, 2000.

[21] K. T. Simons, C. Strauss, and D. BMC Baker. Prospects for ab initio protein structural genomics. 306(5):1191–1199, 2001.

[22] J. Xu and Y. Zhang. How significant is a protein structure similarity with TM-score= 0.5? *Bioinformatics*, 26:889–895, 2010.

[23] A. Roy, A. Kucukural, and Y. Zhang. I-tasser: a unified platform for automated protein structure and function prediction. *Nat. Protoc.*, 5(4):725–738, 2010.

[24] Y. Zhang. I-TASSER server for protein 3D structure prediction. *BMC bioinformatics*, 9:40, 2008.

[25] P. Bradley, D. Chivian, J. Meiler, K. Misura, C. A. Rohl, W. R. Schief, W. J. Wedemeyer, O. Schueler-Furman, P. Murphy, J. Schonbrun, et al. Rosetta predictions in casp5: successes, failures, and prospects for complete automation. *Proteins: Struct. Funct. Bioinf.*, 53(S6):457–468, 2003.

[26] C. Summa and M. Levitt. Near-native structure refinement using *in vacuo* energy minimization. *Proc. Natl. Acad. Sci. USA*, 104:3177–3182, 2007.

[27] G. Chopra, N. Kalisman, and M. Levitt. Consistent refinement of submitted models at CASP using a knowledge-based potential. *Proteins: Struct. Funct. Bioinf.*, 78:2668–2678, 2010.

[28] D. Bhattachary and J. Cheng. 3Drefine: consistent protein structure refinement by optimizing hydrogen bonding network and atomic level refinement. *Proteins: Struct. Funct. Bioinf.*, 81:119–131, 2013.

[29] A. Raval, S. Piana, M. P. Eastwood, R. O. Dror, and D. E. Shaw. Refinement of protein structure homology models via long, all-atom molecular dynamics simulations. *Proteins: Struct. Funct. Bioinf.*, 80(8):2071–2079, 2012.

[30] V. Mirjalili and M. Feig. Protein structure refinement through structure selection and averaging from molecular dynamics ensembles. *J. Chem. Theory Comput.*, 9(2):1294–1303, 2013.

[31] V. Mirjalili, K. Noyes, and M. Feig. Physics based protein structure refinement through multiple molecular dynamics trajectories and structure averaging. *Proteins: Struct. Funct. Bioinf.,*, 2013.

[32] A. W. Stumpff-Kane, K. Maksimiak, M. S. Lee, and M. Feig. Sampling of near-native protein conformations during protein structure refinement using a coarse-grained model, normal modes, and molecular dynamics simulations. *Proteins: Struct. Funct. Bioinf.*, 70(4):1345–1356, 2008.

[33] P. Gniewek, A. Kolinski, R. L. Jernigan, and A. Kloczkowski. How noise in force fields can affect the structural refinement of protein models? *Proteins: Struct. Funct. Bioinf.*, 80(2):335–341, 2012.

[34] C. De Boor et al. A practical guide to splines. 1978.

[35] M. J. Sippl and S. Weitckus. Detection of native-like models for amino acid sequences of unknown three-dimensional structure in a data base of known protein conformations. *Proteins: Struct. Funct. Bioinf.,*, 13(3):258–271, 1992.

[36] S. Miyazawa and R. L. Jernigan. Residue-residue potentials with a favorable contact pair term and an unfavorable high packing density term, for simulation and threading. *J. Mol. Biol.*, 256:623–644, 1996.

[37] A. Kolinski, W. Galazka, and J. Skolnick. On the origin of the cooperativity of protein folding: implications from model simulations. *Proteins: Struct. Funct. Bioinf.,*, 26(3):271–287, 1996.

[38] B. Lee and F. M. Richards. The interpretation of protein structures: estimation of static accessibility. *J. Mol. Biol.*, 55(3):379–IN4, 1971.

[39] D. Eisenberg and A. D. McLachlan. Solvation energy in protein folding and binding. 1986.

[40] K. Lindorff-Larsen, P. Røgen, E. Paci, M. Vendruscolo, and C. M. Dobson. Protein folding and the organization of the protein topology universe. *trends biochem. sci.*, 30(1):13–19, 2005.

[41] P. Røgen and P. Karlsson. Quantifying the relative positions of proteins secondary structure elements. *Geometrica Dedicata*, 134:91–107, 2008.

[42] D. Tobi and R. Elber. Distance-dependent, pair potential for protein folding: Results from linear optimization. *Proteins: Struct. Funct. Bioinf.*, 41:40–46, 2000.

[43] D. Tobi, G. Shafran, N. Linial, and R. Elber. On the design and analysis of protein folding potentials. *Proteins: Struct. Funct. Bioinf.*, 40:71–85, 2000.

[44] J. Meller, M. Wagner, and R. Elber. Maximum feasibility guideline in the design and analysis of protein folding potentials. *J. Comput. Chem.*, 23(1):111–118, 2002.

[45] M. Wagner, J. Meller, and R. Elber. Large-scale linear programming techniques for the design of protein folding potentials. *Mathematical Programming*, 101:301–318, 2004.

[46] J. Qiu and R. Elber. Atomically detailed potentials to recognize native and approximate protein structures. *Proteins: Struct. Funct. Bioinf.,*, P61(1):44–55, 2005.

[47] R. Rajgaria, S. McAllister, and C. Floudas. A novel high resolution Cα–Cα distance dependent force field based on a high quality decoy set. *Proteins: Struct. Funct. Bioinf.*, 65:726–741, 2006.

[48] M. Vendruscolo, R. Najmanovich, and E. Domany. Can a pairwise contact potential stabilize native protein folds against decoys obtained by threading? *Proteins: Struct. Funct. Bioinf.*, 38(2):134–148, 2000.

[49] J. U. Bowie, R. Luthy, and D. Eisenberg. A method to identify protein sequences that fold into a known three-dimensional structure. *Science*, 253(5016):164–170, 1991.

[50] V. N. Maiorov and G. M. Grippen. Contact potential that recognizes the correct folding of globular proteins. *J. Mol. Biol.*, 227(3):876–888, 1992.

[51] R. A. Goldstein, Z. A. Luthey-Schulten, and P. G. Wolynes. Optimal protein-folding codes from spin-glass theory. *Proc. Natl. Acad. Sci. USA*, 89(11):4918–4922, 1992.

[52] L. Mirny and E. Shakhnovich. How to derive a protein folding potential? a new approach to an old problem. *J. Mol. Biol.*, 264:1164–1179, 1996.

[53] Y. Fujitsuka, S. Takada, Z. A. Luthey-Schulten, and P. G. Wolynes. Optimizing physical energy functions for protein folding. *Proteins: Struct. Funct. Bioinf.,*, 54(1):88–103, 2004.

[54] U. Bastolla, J. Farwer, E. W. Knapp, and M. Vendruscolo. How to guarantee optimal stability for most representative structures in the protein data bank. *Proteins: Struct. Funct. Bioinf.*, 44(2):79–96, 2001.

[55] Y. Zhang, A. Kolinski, and J. Skolnick. TOUCHSTONE II: a new approach to ab initio protein structure prediction. *Biophys. J.*, 85:1145, 2003.

[56] Y. Zhang and J. Skolnick. Automated structure prediction of weakly homologous proteins on a genomic scale. *Proc. Natl. Acad. Sci. USA*, 101:7594–7599, 2004.

[57] B. Fain and M. Levitt. Funnel sculpting for in silico assembly of secondary structure elements of proteins. *Proc. Natl. Acad. Sci. USA*, 100:10700–10705, 2003.

[58] J. J. Moré and D. C. Sorensen. On the use of directions of negative curvature in a modified newton method. *Math. Prog.*, 16(1):1–20, 1979.

[59] A. Forsgren and U. Ringertz. On the use of a modified newton method for nonlinear finite element analysis. *Comput. Method. Appl. M.*, 110(3):275–283, 1993.

[60] H. B. Schlegel. Geometry optimization. *Wiley Interdiscip. Rev.: Comput. Mol. Sci.*, 1(5):790–809, 2011.

[61] N. Go, T. Noguti, and T. Nishikawa. Dynamics of a small globular protein in terms of low-frequency vibrational modes. *Proc. Natl. Acad. Sci. USA*, 80(12):3696–3700, 1983.

[62] K. Lindorff-Larsen, N. Trbovic, P. Maragakis, S. Piana, and D. E. Shaw. Structure and dynamics of an unfolded protein examined by molecular dynamics simulation. *J. Am. Chem. Soc.*, 134(8):3787–3791, 2012.

[63] V. V. Rybkin, U. Ekström, and T. Helgaker. Internal-to-cartesian back transformation of molecular geometry steps using high-order geometric derivatives. *J. Comput. Chem.*, 34:1842–1849, 2013.

[64] E. B. Wilson, J. C. Decius, and P. C. Cross. *Molecular Vibrations*. McGraw-Hill, New York, 1955.

[65] T. Noguti and N. Go. A method of rapid calculation of a second derivative matrix of conformational energy for large molecules. *J. Phys. Soc. Jpn.*, 52(10):3685–3690, 1983.

[66] S. Nakamura, M. Ikeguchi, and K. Shimizu. Parallel algorithm for efficient calculation of second derivatives of conformational energy function in internal coordinates. *J. Comput. Chem.*, 19(15):1716–1723, 1998.

[67] K. Kamiya, Y. Sugawara, and H. Umeyama. Algorithm for normal mode analysis with general internal coordinates. *J. Comput. Chem.*, 24(7):826–841, 2003.

[68] C. Peng, P. Y. Ayala, H. B. Schlegel, and M. J. Frisch. Using redundant internal coordinates to optimize equilibrium geometries and transition states. *J. Comput. Chem.*, 17(1):49–56, 1996.

[69] P. Pulay, G. Fogarasi, F. Pang, and J. E. Boggs. Systematic ab initio gradient calculation of molecular geometries, force constants, and dipole moment derivatives. *J. Am. Chem. Soc.*, 101(10):2550–2560, 1979.

[70] M. Bixon and S. Lifson. Potential functions and conformations in cycloalkanes. *Tetrahedron*, 23(2):769–784, 1967.

[71] R. F. Hout, B. A. Levi, and W. J. Hehre. A method for the calculation of normal-mode vibrational frequencies using symmetry coordinates. application to the calculation of secondary deuterium isotope effects on carbocations. *J. Comput. Chem.*, 4(4):499–505, 1983.

[72] H. Bekker, H. Berendsen, and W. F. van Gunsteren. Force and virial of torsional-angle-dependent potentials. *J. Comput. Chem.*, 16(5):527–533, 1995.

[73] A. Blondel and M. Karplus. New formulation for derivatives of torsion angles and improper torsion angles in molecular mechanics: Elimination of singularities. *J. Comput. Chem.*, 17(9):1132–1141, 1996.

[74] R. E. Tuzun, D. W. Noid, and B. G. Sumpter. Computation of internal coordinates, derivatives, and gradient expressions: torsion and improper torsion. *J. Comput. Chem.*, 21(7):553–561, 2000.

[75] G. Fogarasi, X. Zhou, P. W. Taylor, and P. Pulay. The calculation of ab initio molecular geometries: efficient optimization by natural internal coordinates and empirical correction by offset forces. *J. Am. Chem. Soc.*, 114(21):8191–8201, 1992.

[76] H. Eyring. The resultant electric moment of complex molecules. *Phys. Rev. Lett.*, 39(4):746, 1932.

[77] J. Pesonen. Vibrational coordinates and their gradients: A geometric algebra approach. *J. Chem. Phys.*, 112:3121, 2000.

[78] K. J. Miller, R. J. Hinde, and J. Anderson. First and second derivative matrix elements for the stretching, bending, and torsional energy. *J. Comput. Chem.*, 10(1):63–76, 1989.

[79] S. J. Leon. *Linear algebra with applications*. Pearson Education, 2007.

[80] M. Levitt, M. Hirshberg, R. Sharon, and V. Daggett. Potential energy function and parameters for simulations of the molecular dynamics of proteins and nucleic acids in solution. *Comput. Phys. Commun.*, 91(1):215–231, 1995.

[81] J. Moult, K. Fidelis, A. Kryshtafovych, and A. Tramontano. Critical assessment of methods of protein structure prediction (CASP)-round IX. *Proteins: Struct. Funct. Bioinf.*, 79:1–5, 2011.

[82] D. Cozzetto, A. Kryshtafovych, and A. Tramontano. Evaluation of CASP8 model quality predictions. *Proteins: Struct. Funct. Bioinf.*, 77:157–166, 2009.

[83] A. Kryshtafovych, K. Fidelis, and A. Tramontano. Evaluation of model quality predictions in CASP9. *Proteins: Struct. Funct. Bioinf.*, 79:91–106, 2011.

[84] T. Lazaridis and M. Karplus. Effective energy functions for protein structure prediction. *Curr. Opin. Struct. Biol.*, 10:139–145, 2000.

[85] H. Zhou and J. Skolnick. GOAP: a generalized orientation-dependent, all-atom statistical potential for protein structure prediction. *Biophys. J.*, 101:2043–2052, 2011.

[86] J. Skolnick. In quest of an empirical potential for protein structure prediction. *Curr. Opin. Struct. Biol.*, 16:166–171, 2006.

[87] J. Zhu, H. Fan, X. Peiole, B. Honig, and A. Mark. Refining homology models by combining replica-exchange molecular dynamics and statistical potentials. *Proteins: Struct. Funct. Bioinf.*, 72:1171–1188, 2008.

[88] Y. Amautova and H. Scheraga. Use of decoys to optimize an all-atom forcefield including hydration. *Biophys. J.*, 95:2434–2449, 2008.

[89] C. Rohl, C. Strauss, K. Misura, and D. Baker. Protein structure prediction using Rosetta. *Methods Enzymol.*, 383:66–93, 2004.

[90] Y. Zhang, A. Kolinski, and J. Skolnick. Touchstone II: A new approach to ab initio protein structure prediction. *Biophys. J.*, 85:1145–1164, 2003.

[91] P. Benkert, S. Tosatto, and D. Schomburg. QMEAN: A comprehensive scoring function for model quality assessment. *Proteins: Struct. Funct. Bioinf.*, 71:261–277, 2008.

[92] Y. Zhang and J. Skolnick. Automated structure prediction of weakly homologous proteins on a genomic scale. *Proc. Natl. Acad. Sci. USA*, 101:7594–7599, 2004.

[93] R. Samudrala and M. Levitt. Decoys 'R'Us: A database of incorrect conformations to improve protein structure prediction. *Protein Science*, 9:1399–1401, 2008.

[94] J. Tsai, R. Bonneau, A. Morozov, B. Kuhlman, C. Rohl, and D. Baker. An improved protein decoy set for testing energy functions for protein structure prediction. *Proteins: Struct. Funct. Bioinf.*, 53:76–87, 2003.

[95] A. Kryshtafovych, A. Barbato, K. Fidelis, B. Monastyrskyy, T. Schwede, and A. Tramontano. Assessment of the assessment: evaluation of the model quality estimates in CASP10. *Proteins: Struct. Funct. Bioinf.*, 82:112–126, 2014.

[96] K. Kaindl and B. Steipe. Metric properties of the root-mean square deviation of vector sets. *Acta Cryst. A*, 53:809, 1997.

[97] M. Tirion. Large amplitude elastic motions in proteins from a single-parameter, atomic analysis. *Phys. Rev. Lett.*, 77:1905–1908, 1996.

[98] F. Tama and Y. Sanejouand. Conformational change of proteins arising from normal mode calculations. *Protein Eng.*, 14:1–6, 2001.

[99] J. Bohr, H. Bohr, S. Brunak, R. Cotterill, H. Fredholm, B. Lautrup, and S. Petersen. Protein structures from distance inequalities. *J. Mol. Biol.*, 231:861–869, 1993.

[100] C. Summa and M. Levitt. Near-native structure refinement using in vacuo energy minimization. *Proc. Natl. Acad. Sci. USA*, 104:3177–3182, 2007.

[101] I. Bahar, A. Atilgan, and B. Erman. Direct evaluation of thermal fluctuations in proteins using a single-parameter harmonic potential. *Folding and Design*, 2:173–181, 1997.

[102] A. Atilgan, S. Durell, R. Jernigan, M. Demirel, O. Keskin, and I. Bahar. Anisotropy of fluctuation dynamics of proteins with an elastic network model. *Biophys. J.*, 80:505–515, 2001.

[103] M. Toda. Vibration of a chain with nonlinear interaction. *J. Phys. Soc. Japan*, 22:431–436, 1967.

[104] C. de Boor. *A practical guide to splines*. Springer, 1978.

[105] J. Eickholt, Z. Wang, and J. Cheng. A conformation ensemble approach to protein residue-residue contact. *BMC structural biology*, 11:38, 2011.

[106] J. Handl, J. Knowles, and S. Lovell. Artefacts and biases affecting the evaluation of scoring functions on decoy sets for protein structure prediction. *Bioinformatics*, 25:1271–1279, 2009.

[107] P. Güntert, C. Mumenthaler, and K. Wüthrich. Torsion angle dynamics for NMR structure calculation with the new program DYANA. *J. Mol. Biol.*, 273:283–298, 1997.

[108] B. Park and M. Levitt. Energy functions that discriminate x-ray and near-native folds from well-constructed decoys. *J. Mol. Biol.*, 258:367–392, 1996.

[109] K. Simons, C. Kooperberg, E. Huang, and D. Baker. Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and bayesian scoring functions. *J. Mol. Biol.*, 268:209–225, 1997.

[110] C. Keasar and M. Levitt. A novel approach to decoy set generation: designing a physical energy function having local minima with native structure characteristics. *J. Mol. Biol.*, 329:159–174, 2003.

[111] E. Huang. A combined approach for ab initio construction of low resolution protein tertiary structures from sequence. In *Pacific Symposium on Biocomputing*, volume 4, pages 505–516, 1999.

[112] Y. Xia, E. Huang, M. Levitt, and R. Samudrala. Ab initio construction of protein tertiary structures using a hierarchical approach. *J. Mol. Biol.*, 300:171–185, 2000.

[113] K. Simons, I. Ruczinski, C. Kooperberg, B. Fox, C. Bystroff, and D. Baker. Improved recognition of native-like protein structures using a combination of sequence-dependent and sequence-independent features of proteins. *Proteins: Struct. Funct. Bioinf.*, 34:82–95, 1999.

[114] J. Zhang and Y. Zhang. A novel side-chain orientation dependent potential derived from random-walk reference state for protein fold selection and structure prediction. *PloS One*, 5:e15386, 2010.

[115] Y. Zhang and J. Skolnick. Scoring function for automated assessment of protein structure template quality. *Proteins: Struct. Funct. Bioinf.*, 57:702–710, 2004.

[116] R. Samudrala and J. Moult. An all-atom distance-dependent conditional probability discriminatory function for protein structure prediction. *J. Mol. Biol.*, 275:895–916, 1998.

[117] K. Lindorff-Larsen, S. Piana, K. Palmo, P. Maragakis, J. L. Klepeis, R. O. Dror, and D. E. Shaw. Improved side-chain torsion potentials for the Amber ff99SB protein force field. *Proteins: Struct. Funct. Bioinf.*, 78:1950–1958, 2010.

[118] H. Zhou and Y. Zhou. Distance-scaled, finite ideal-gas reference state improves structure-derived potentials of mean force for structure selection and stability prediction. *Protein Sci.*, 11:2714–2726, 2002.

[119] K. Lindorff-Larsen, S. Piana, R. O. Dror, and D. E. Shaw. How fast-folding proteins fold. *Science*, 334(6055):517–520, 2011.

[120] M. Carlsen, P. Koehl, and P. Røgen. On the importance of the distance measures used to train and test knowledge-based potentials for proteins. *PLOS ONE*, 9(11):e109335, 2014.

[121] B. Fain, Y. Xia, and M. Levitt. Design of an optimal chebyshev-expanded discrimination function for globular proteins. *Protein science*, 11(8):2010–2021, 2002.

[122] A. Liwo, M. R. Pincus, R. J. Wawak, S. Rackovsky, S. Ołdziej, and H. A. Scheraga. A united-residue force field for off-lattice protein-structure simulations. ii. parameterization of short-range interactions and determination of weights of energy terms by z-score optimization. *J. Comput. Chem.*, 18(7):874–887, 1997.

[123] A. Liwo, S. Oldziej, R. Kazmierkiewicz, M. Groth, and C. Czaplewski. Design of a knowledge-based force field for off-lattice simulations of protein structure. *Acta Biochim. Pol.*, 44:527–548, 1997.

[124] K. Lum, D. Chandler, and J. D. Weeks. Hydrophobicity at small and large length scales. *J. Phys. Chem. B.*, 103(22):4570–4577, 1999.

[125] M. Carlsen. Using operators to expand the block matrices forming the hessian of a molecular potential. *J. Comput. Chem.*, 35(15):1149–1158, 2014.

[126] A. Schug and W. Wenzel. An evolutionary strategy for all-atom folding of the 60-amino-acid bacterial ribosomal protein l20. *Biophys. J.*, 90(12):4273–4280, 2006.

[127] K. A. Dill and J. L. MacCallum. The protein-folding problem, 50 years on. *Science*, 338(6110):1042–1046, 2012.

[128] K. Dill and H. Chan. From levinthal to pathways to funnels. *Nature Struct. Biol.*, 4:10–19, 1997.

[129] L. Piela, J. Kostrowicki, and H. A. Scheraga. On the multiple-minima problem in the conformational analysis of molecules: deformation of the potential energy hypersurface by the diffusion equation method. *J. Phys. Chem.*, 93(8):3339–3346, 1989.

[130] D. J. Wales and H. A. Scheraga. Global optimization of clusters, crystals, and biomolecules. *Science*, 285(5432):1368–1372, 1999.

[131] G. M. Crippen. Prediction of protein folding from amino acid sequence over discrete conformation spaces. *Biochemistry*, 30(17):4232–4237, 1991.

[132] C. Loose, J. Klepeis, and C. Floudas. A new pairwise folding potential based on improved decoy generation and side-chain packing. *Proteins: Struct. Funct. Bioinf.*, 54(2):303–314, 2004.

[133] R. Rajgaria, S. McAllister, and C. Floudas. Distance dependent centroid to centroid force fields using high resolution decoys. *Proteins: Struct. Funct. Bioinf.*, 70(3):950–970, 2008.

[134] J. Pillardy, C. Czaplewski, A. Liwo, J. Lee, D. R. Ripoll, R. Kaźmierkiewicz, S. Ołdziej, W. J. Wedemeyer, K. D. Gibson, Y. A. Arnautova, et al. Recent improvements in prediction of protein structure by global optimization of a potential energy function. *Proc. Natl. Acad. Sci. USA*, 98(5):2329–2333, 2001.

[135] H. Lu and J. Skolnick. A distance-dependent atomic knowledge-based potential for improved protein structure selection. *Proteins: Struct. Funct. Bioinf.*, 44(3):223–232, 2001.

[136] C. Floudas, H. Fung, S. McAllister, M. Mönnigmann, and R. Rajgaria. Advances in protein structure prediction and de novo protein design: A review. *Chem. Eng. Sci.*, 61(3):966–988, 2006.

[137] J. Bryngelson and P. Wolynes. Spin glasses and the statistical mechanics of protein folding. *Proc. Natl. Acad. Sci. USA*, 84:7524–7528, 1987.

[138] P. Leopold, M. Montal, and J. Onuchic. Protein folding funnels: a kinetic approach to the sequence-structure relationship. *Proc. Natl. Acad. Sci. USA*, 89:8721–8725, 1992.

[139] I. Iben, D. Braunstein, W. Doster, H. Frauenfelder, M. Hong, J. Johnson, S. Luck, P. Ormos, A. Schulte, P. Steinbach, et al. Glassy behavior of a protein. *Phys. Rev. Lett.*, 62(16):1916–1919, 1989.

[140] P. G. Wolynes, J. N. Onuchic, and D. Thirumalai. Navigating the folding routes. *Science*, 267(5204):1619–1620, 1995.

[141] P. Røgen and B. Fain. Automatic classification of protein structure by using gauss integrals. *Proc. Natl. Acad. Sci. USA*, 100(1):119–124, 2003.

[142] M. Carlsen and P. Røgen. Protein structure refinement by optimization. *Proteins: Struct. Funct. Bioinf.*, 2015.

[143] M. Yamashita, K. Fujisawa, and M. Kojima. Sdpara: Semidefinite programming algorithm parallel version. *Parallel Comput.*, 29(8):1053–1067, 2003.

[144] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM review*, 38(1):49–95, 1996.

[145] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.